

Pengenalan Aksara Pallawa dengan Model Hidden Markov

Wiwien Widyastuti

Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Sanata Dharma

Email: wiwien@usd.ac.id

Abstrak

Aksara Pallawa atau kadangkala ditulis sebagai Pallava adalah sebuah aksara yang berasal dari India bagian selatan. Aksara ini sangat penting untuk sejarah di Indonesia karena aksara ini merupakan aksara dari mana aksara-aksara Nusantara diturunkan [Wikipedia]. Hidden Markov Model (HMM) adalah suatu metode stochastic yang sudah banyak digunakan pada sistem pengenalan suara dan sistem pengenalan pola dan menghasilkan tingkat pengenalan yang cukup tinggi [Intechweb, 2011]. Penelitian ini menerapkan dan mengamati unjuk kerja Model Hidden Markov untuk mengenali aksara Pallawa. Kegiatan penelitian meliputi pengambilan data baik untuk pelatihan maupun untuk pengujian, preprocessing, ekstraksi ciri, tahap pelatihan untuk mencari model Hidden Markovnya terakhir adalah tahap pengujian serta pembahasan hasil penelitian. Berdasarkan hasil pengamatan dengan variasi jumlah *state* sebesar 8, 10, 12, 14, 16, 18, dan 20 diperoleh hasil pengenalan terbaik pada jumlah *state* 14 yaitu sebesar 35,1515% dengan waktu eksekusi selama 82 detik.

Kata Kunci: Aksara Pallawa, Hidden Markov Model.

1. Pendahuluan

Pada masa kini Aksara Nusantara merupakan salah satu warisan budaya yang nyaris punah. Tentunya hal ini sangat disayangkan mengingat betapa berharganya warisan budaya tersebut. Hampir semua aksara daerah di Indonesia merupakan turunan dari Aksara Pallawa yang berasal dari daerah India Selatan. Banyak prasasti di Nusantara yang ditulis menggunakan aksara Pallawa sehingga aksara ini sangat penting untuk sejarah di Indonesia [Wikipedia]. Aksara Pallawa dapat dilihat pada gambar 1.



Gambar 1. Aksara-Aksara Palawa[Wikipedia]

Sejalan dengan perkembangan komputer modern yang sangat pesat dengan

kemampuan komputasi yang semakin tinggi, semakin banyak masalah yang dapat dipecahkan dengan bantuan komputasi menggunakan komputer. Model komputasi *machine learning* seperti Hidden Markov Model juga dapat memanfaatkan kemampuan komputer untuk menangani masalah-masalah yang kompleks di antaranya adalah klasifikasi pola, pengenalan pola, pengenalan wajah, pencitraan medis, kendali, prediksi saham dan lain sebagainya [Ganeshamoorthy, 2008].

Penggunaan Model Hidden Markov untuk pengenalan pola telah banyak dilakukan oleh para peneliti. Anastasia Rita Widiarti telah menggunakan Hidden Markov Model (HMM) untuk mengenali karakter Jawa dengan akurasi 85,7% [Widiarti, 2009]. Pengenalan aksara Cina dengan HMM juga telah dilakukan Bing Feng dan hasilnya menyatakan bahwa HMM merupakan pendekatan yang menjanjikan dalam mengenali aksara Cina[Feng, 2002]. Sherif Abdou menggunakan HMM untuk mengenali tulisan tangan Arab [Hosny, 2011]. Semua hasil penelitian tersebut menunjukkan hasil yang baik

Penelitian ini akan mencoba menerapkan Model Hidden Markov untuk mengenali aksara Pallawa dan melihat bagaimana unjuk kerjanya.

Model Hidden Markov merupakan *finite learnable stochastic automate*, yang dapat dirangkum sebagai proses stochastic ganda dengan dua aspek berikut [Intechweb, 2011]:

1. Proses stochastic pertama adalah suatu himpunan berhingga keadaan yang masing-

masing berhubungan dengan distribusi probabilitas dimensional. Transisi antara keadaan yang berbeda diatur secara statistik oleh probabilitas-probabilitas yang disebut probabilitas transisi.

2. Proses stochastic yang kedua, di keadaan mana suatu kejadian dapat diobservasi. Karena kita hanya menganalisa sesuatu yang diobservasi tanpa melihat keadaan mana yang terjadi, maka keadaan tersebut "hidden" bagi peneliti, sehingga dinamakan "Hidden Markov model".

Setiap model Hidden Markov ditentukan oleh keadaan, probabilitas keadaan, probabilitas transisi, probabilitas emisi dan probabilitas inisial.

Untuk mendefinisikan HMM secara lengkap, harus ditentukan lima elemen berikut [Intechweb, 2011] :

1. N keadaan dari model, yang ditentukan oleh : $S = \{S_1, \dots, S_N\}$
2. M symbol observasi tiap keadaan $V = \{v_1, \dots, v_M\}$. Jika penelitiannya kontinyu mana M tidak berhingga.
3. Distribusi peluang transisi keadaan $A = \{a_{ij}\}$, dengan a_{ij} adalah probabilitas bahwa keadaan pada saat t+1 adalah S_j , diberikan saat keadaan saat t adalah S_i . Struktur dari matriks stochastic menentukan struktur koneksi dari model. Jika koefisien a_{ij} nol, maka akan tetap nol walaupun melewati proses pelatihan, sehingga tidak akan pernah ada transisi dari keadaan S_i ke

Dengan q_i , melambangkan keadaan sekarang. Probabilitas transisi seharusnya memenuhi kekangan

$$S_j \cdot a_{ij} = p\{q_{t+1} = j | q_t = i\}, 1 \leq i, j \leq N$$

stochastic normal : $a_{ij} \geq 0, 1 \leq i, j \leq N$

$$\text{dan } \sum_{j=1}^N a_{ij} = 1, 1 \leq i < N$$

4. Distribusi probabilitas symbol observasi pada tiap keadaan, $B = \{b_j(k)\}$ dengan $b_j(k)$ adalah probabilitas bahwa symbol v_k diemisikan pada keadaan S_j .

$$b_j(k) \geq 0, 1 \leq j \leq N, 1 \leq k \leq M$$

dan $\sum_{k=1}^M b_j(k) = 1, 1 \leq j \leq N$ (2)

Jika observasi kontinyu, harus digunakan *continuous probability density function*. Biasanya probabilitas density diaproksimasi dengan sejumlah M distribusi Gaussian N,

$$b_j(O_t) = \sum_{m=1}^M c_{jm} N(\mu_{jm}, \Sigma_{jm}, O_t) \quad (3)$$

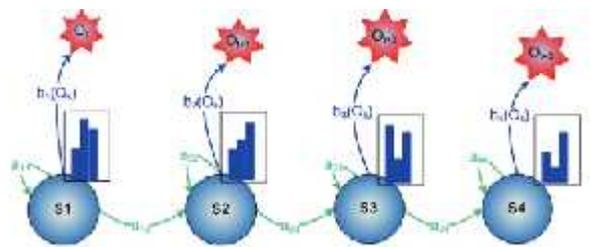
Dengan c_{jm} = koefisien pembobotan,

μ_{jm} = mean vector, Σ_{jm} = matriks kovarian. c_{jm} seharusnya juga memenuhi asumsi stochastic $c_{jm} \geq 0, 1 \leq j \leq N, 1 \leq m \leq M$ dan

$$\sum_{m=1}^M c_{jm} = 1, 1 \leq j \leq N \quad (4)$$

5. HMM adalah distribusi keadaan inisial $f = \{f_i\}$, dengan f_i , adalah probabilitas bahwa model berada pada keadaan S_i saat t=0 dengan

$$f_i = p\{q_1 = i\} \text{ dan } 1 \leq i \leq N \quad (5)$$



Gambar 2. Contoh HMM [Intechweb, 2011]

Tiga Masalah Dasar dari HMM adalah sebagai berikut [Intechweb, 2011]:

1. Masalah evaluasi
 Bagaimanakah probabilitas suatu observasi $O = o_1, o_2, \dots, o_t$ dibangkitkan oleh suatu model $p(O | \lambda)$ dengan HMM ?
2. Masalah *decoding*
 Manakah deretan keadaan yang paling mungkin pada model yang menghasilkan observasi $O = o_1, o_2, \dots, o_t$?
3. Masalah Pelatihan
 Bagaimanakah seharusnya parameter model $\{A, B, \lambda\}$ disesuaikan supaya memaksimalkan $p(O | \lambda)$, jika diberikan model dan deret observasi $O = o_1, o_2, \dots, o_t$?

Masalah Evaluasi dan Algoritma Forward

Diberikan suatu model $\lambda = (A, B, \lambda)$ dan deret observasi $O = o_1, o_2, \dots, o_t$, $p(O | \lambda)$ akan dicari. Metode perhitungan dengan kompleksitas yang rendah menggunakan variable bantu :

$$\Gamma_t(i) = p\{o_1, o_2, \dots, o_t, q_t = i | \lambda\} \quad (6)$$

$\Gamma_t(i)$ disebut variable forward, dan $O = o_1, o_2, \dots, o_t$ adalah deret observasi parsial.

Terdapat suatu hubungan rekursif :

$$r_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^N r_t(i) a_{ij}, 1 \leq j \leq N, 1 \leq t \leq T-1 \quad (7)$$

Dengan $r_1(j) = f_j b_j(o_1), 1 \leq j \leq N$.

$r_T(i), 1 \leq i \leq N$ dapat dihitung dengan rekursi tersebut. Sehingga probabilitas yang dibutuhkan adalah :

$$p\{O|\bar{\}} = \sum_{i=1}^N r_T(i) \quad (8)$$

Metode tersebut dikenal dengan algoritma *forward*.

Variabel *backward* $S_t(i)$ dapat ditentukan dengan cara yang sama

$$S_t(i) = p\{o_{t+1}, o_{t+2}, \dots, o_T | q_t = i, \bar{\}} \quad (9)$$

Jika diberikan keadaan sekarang adalah i , $S_t(i)$ adalah probabilitas dari deret observasi parsial $o_{t+1}, o_{t+2}, \dots, o_T$.

$S_t(i)$ dapat juga dihitung secara efisien menggunakan suatu rekursif

$$S_t(i) = \sum_{j=1}^N S_{t+1}(j) a_{ij} b_j(o_{t+1}), 1 \leq i \leq N, 1 \leq t \leq T-1 \quad (10)$$

Dengan $S_T(i) = 1, 1 \leq i \leq N$

Selanjutnya dapat dilihat,

$$r_t(i) S_t(i) = p\{O, q_t = i | \bar{\}}, 1 \leq i \leq N, 1 \leq t \leq T \quad (11)$$

Sehingga ada dua cara menghitung $p\{O|\bar{\}}$, dengan variable *forward* atau variable *backward* :

$$p\{O|\bar{\}} = \sum_{i=1}^N p\{O, q_t = i | \bar{\}} = \sum_{i=1}^N r_t(i) S_t(i) \quad (12)$$

Masalah Decoding dan Algoritma Viterbi

Diberikan suatu model $\lambda = (A, B, \pi)$ dan deret observasi $O = o_1, o_2, \dots, o_T$ akan dicari deret keadaan yang paling mungkin. Definisi dari "kemungkinan deret keadaan" mempengaruhi solusi permasalahan ini. Untuk memecahkan permasalahan ini digunakan algoritma Viterbi untuk menemukan semua deret keadaan dengan *maximum likelihood*.

Sebuah variabel bantu ditentukan sedemikian hingga probabilitas tertinggi dari deret observasi parsial dan deret keadaan sampai pada $t=t$, jika diberikan keadaan saat ini adalah i .

$$u_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} p\{q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_{t-1} | \bar{\}} \quad (13)$$

Maka

$$u_{t+1}(j) = b_j(o_{t+1}) \left[\max_{1 \leq i \leq N} u_t(i) a_{ij} \right], 1 \leq j \leq N, 1 \leq t \leq T-1 \quad (14)$$

Dengan $u_1(j) = f_j b_j(o_1), 1 \leq j \leq N$

Sehingga dimulai dari perhitungan $u_T(j), 1 \leq j \leq N$ untuk menghitung deret keadaan yang paling mungkin.

Masalah Pelatihan dan Algoritma Baum-Welch

Algoritma Baum-Welch juga dikenal dengan algoritma Forward-Backward. Metode ini menggunakan kalkulus untuk untuk memaksimalkan kuantitas bantuan.

$$Q(\lambda, \bar{\lambda}) = \sum_q p\{q | O, \lambda\} \log [p\{O, q, \bar{\lambda}\}] \quad (15)$$

Sebagai tambahan untuk variabel *forward* dan *backward*, diperlukan dua variabel tambahan lagi. Variabel pertama adalah :

$$\langle_t(i, j) = p\{q_t = i, q_{t+1} = j | O, \bar{\}} \quad (16)$$

Yang bisa ditulis juga sebagai berikut :

$$\langle_t(i, j) = \frac{p\{q_t = i, q_{t+1} = j, O | \bar{\}}}{p\{O | \bar{\}}} \quad (17)$$

Dapat digunakan variabel *forward* dan *backward* dan hasil pada :

$$\langle_t(i, j) = \frac{r_t(i) a_{ij} S_{t+1}(j) b_j(o_{t+1})}{\sum_{i=1}^N \sum_{j=1}^N r_t(i) a_{ij} S_{t+1}(j) b_j(o_{t+1})} \quad (18)$$

Variabel kedua adalah probabilitas *posteriori*.

$$x_t(i) = \left[\frac{r_t(i) S_t(i)}{\sum_{i=1}^N r_t(i) S_t(i)} \right] \quad (19)$$

Sehingga hubungan antara $x_t(i)$ dan $\langle_t(i, j)$ adalah :

$$x_t(i) = \sum_{j=1}^N \langle_t(i, j), 1 \leq i \leq N, 1 \leq t \leq M \quad (20)$$

Untuk memaksimalkan kuantitas $p\{O|\bar{\}}$, sekarang bisa dideskripsikan proses pelatihan Baum-Welch. Asumsikan model mula-mula $\lambda = (A, B, \pi)$ dan hitung $Q(\lambda, \bar{\lambda})$ dan λ . Setelah itu hitung λ dan $\bar{\lambda}$. Persamaan berikut dikenal

dengan formula *re-estimasi* yang digunakan untuk memperbarui parameter HMM.

$$\bar{f}_i = x_1(i), 1 \leq i \leq N \quad (21)$$

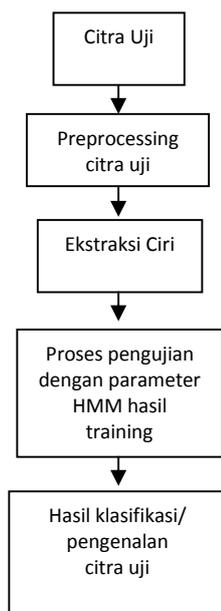
$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} x_t(i, j)}{\sum_{t=1}^{T-1} x_t(i)}, 1 \leq i \leq N, 1 \leq j \leq N \quad (22)$$

2. Metode



Gambar 3. Diagram Alir Tahap Pelatihan

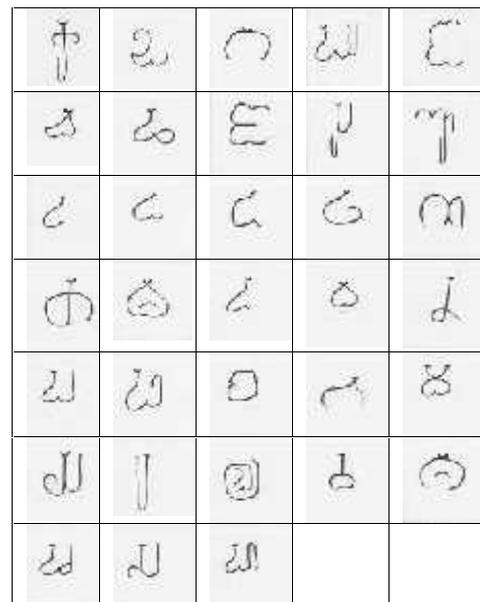
Blok diagram sistem pengenalan yang dikerjakan dalam penelitian terdiri dari dua tahap yaitu tahap pelatihan dan tahap pengujian. Diagram alir tahap pelatihan dapat dilihat pada gambar 3. Diagram alir tahap pengujian dapat dilihat pada gambar 4.



Gambar 4. Diagram Alir Tahap Pengujian

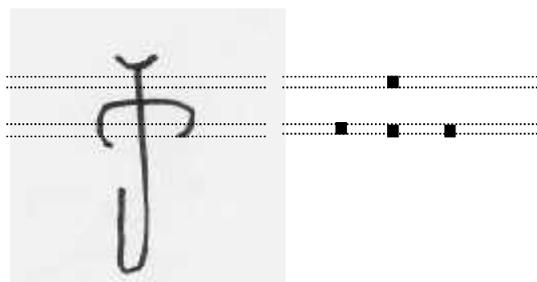
Tahap Preprocessing dan Ekstraksi Ciri Citra Pelatihan

Citra tulisan tangan aksara Pallawa secara keseluruhan berjumlah 33. Setiap aksara Pallawa ditulis sebanyak 20 kali. Kemudian tulisan tersebut di-*scan* dengan format jpg dengan ukuran 64 x 64 pixel. Sehingga diperoleh sebanyak 33 x 20 = 660 citra. Sebanyak 15 x 33 citra digunakan untuk proses pelatihan HMM sedangkan 5 x 33 citra digunakan sebagai citra uji. Citra diubah dalam format *black and white*. Contoh citra hasil scan dapat dilihat pada gambar 5.



Gambar 5. Aksara Palawa hasil rezising dan segmentasi

Pelatihan HMM yang akan digunakan adalah model 1-dimensi diskret, sehingga citra dalam bentuk 2 dimensi selanjutnya diserialkan baris demi baris membentuk sebuah deretan 1 dimensi. Proses selanjutnya adalah ekstraksi ciri. Citra *black and white* yang sudah diserialkan kemudian diekstraksi dengan cara menghitung jumlah *black pixel* di sejumlah baris tertentu. Penelitian ini akan mencoba mengekstrak jumlah *black pixel* pada baris tunggal. Contoh citra hasil scan beserta cara membuat serial data secara vertikal dapat dilihat pada gambar 6.



Gambar 6. Ekstraksi Ciri aksara Palawa 'ka'

Setelah itu citra dipisahkan menjadi 2 bagian, bagian pertama sebanyak 15 citra digunakan sebagai data latihan dan bagian kedua sebanyak 5 citra digunakan sebagai data pengujian.

Tahap Pelatihan

Algoritma pelatihan yang digunakan adalah algoritma Baum-Welch. Algoritma Baum-Welch merupakan prosedur iterative untuk menentukan optimum local. Penentuan optimum local diawali dengan suatu model awal (*initial model*) dari parameter model $\theta = (f, A, B)$. Kemudian dilakukan perhitungan iterative untuk re-estimasi parameter HMM. Algoritma yang digunakan seperti yang dijelaskan pada dasar teori.

Tahap Pengujian

Setelah memperoleh parameter HMM yang merupakan optimum local, tahap selanjutnya adalah mengklasifikasi atau mengenali citra uji. Algoritma yang digunakan adalah algoritma Viterbi.

2.1 Metode Pengumpulan Data

Pada tahap pelatihan dan pengujian, digunakan program HMM toolbox yang dikembangkan oleh Kevin Murphy, 2005 [Murphy]. HMM toolbox mempunyai 3 fungsi utama yang digunakan untuk:

1. Estimasi parameter likelihood maksimum menggunakan algoritma Baum Welch.
2. Klasifikasi Deret, digunakan untuk mengevaluasi *log-likelihood* dari model yang sudah dilatih dengan data pengujian.
3. Menghitung deret dengan probabilitas tertinggi dengan algoritma Viterbi.

Penelitian ini menggunakan dua bagian dari HMM toolbox tersebut yaitu bagian a untuk tahap pelatihan dan bagian b untuk tahap pengujian.

Tahap pelatihan dan pengujian dilakukan dengan mengubah-ubah jumlah *state* pada model Hidden Markov. Jumlah *state* yang digunakan adalah 8, 10, 12, 16, 18, 20. Tahap pelatihan menggunakan 15 data citra untuk masing-masing aksara Palawa. Tahap pengujian menggunakan 5 data citra untuk masing-masing aksara Palawa.

2.2 Metode Analisis Data

Sistem diuji menggunakan 5 kali 5 jenis citra. Masing-masing diuji dengan jumlah *state* 8, 10, 12, 14, 16, 18 dan 20. Hasil dari pengujian kemudian dituliskan dalam bentuk *confusion matrix*. Berdasarkan *confusion matrix* tersebut

dapat diperoleh jumlah pengenalan yang salah dan jumlah pengenalan yang benar pada tiap *state*.

3. Hasil dan Pembahasan

Hasil pengujian secara lengkap dengan menggunakan jumlah *state* 8, 10, 12, 14, 16, 18 dan 20 dapat dilihat pada *confusion matrix*. Selain mengamati tingkat keberhasilan pengenalan, diamati pula waktu yang digunakan untuk mengeksekusi program. Berdasarkan *confusion matrix*, hasil pengamatan dapat dirangkum dalam tabel 1.

Tabel 1. Hasil Pengamatan

No.	Jumlah State	Jumlah benar	Jumlah salah	Prosentase keberhasilan (%)	Waktu yang diperlukan
1.	8	46	119	27,8788	53
2.	10	51	114	30,9091	57
3.	12	54	111	32,7273	72
4.	14	58	107	35,1515	82
5.	16	53	112	32,1212	66
6.	18	51	114	30,9091	87
7.	20	58	107	35,1515	91

Berdasar hasil pengamatan tersebut, tampak bahwa hasil pengenalan terbaik adalah 35,1515% menggunakan jumlah *state* 14, dengan waktu eksekusi 82 detik. Tampak juga bahwa secara keseluruhan hasil pengenalan belum memuaskan bahkan masih jauh dari yang diharapkan. Sehingga diperlukan penelitian lanjutan dengan menggunakan metode ekstraksi ciri yang bermacam-macam serta jumlah *state* juga bisa dibuat lebih bervariasi dan ditingkatkan.

4. Kesimpulan

Berdasarkan hasil pengamatan, dapat dibuat kesimpulan bahwa program pengenalan aksara Palawa menggunakan model Hidden Markov telah berhasil dibuat.

Hasil pengenalan terbaik diperoleh pada model Hidden Markov dengan jumlah *state* 14 yaitu sebesar 35,1515% dengan waktu eksekusi selama 82 detik.

Daftar Pustaka

- Aksara Pallawa. www.Wikipedia.org/wiki/Aksara_Pallawa
- Intechweb.org, (2011). Hidden Markov Models, Theory and Application, Intech, Croatia.

Widiarti, Anastasia Rita. Wastu, Phalita Nari
(2009) Javanese Character Recognition
Using Hidden Markov Model, World
Academy of Science, Engineering and
Technology Vol 3 2009-09-24.

Ganeshamoorthy K, 2008, On the Performance
of Parallel Neural Network
Implementations on Distributed Memory
Architectures, Proceeding of 8th IEEE
International Symposium on Cluster
Computing and the Grid.

Intechweb, 2011Hosny. I, Abdou S, Fahmy A
(2011). Using Advanced Hidden Markov
Models for Online Arabic Handwriting
Recognition, Pattern Recognition (ACPR),
IEEE.

[http://www.cs.ubc.ca/~murphyk/Software/HMM/
hmm.html](http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html)