

Klasifikasi Kesahihan Hadits Berdasarkan Perawi Hadits Menggunakan *Principal Component Analysis* (PCA) dan *Backpropagation Neural Network* (BPNN)

Ulin Nuha¹, Naim Rochmawati²,

¹ Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

² Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya

ulinnuha87@outlook.com

naimrochmawati@unesa.ac.id

Abstrak—Hadits merupakan sumber hukum kedua bagi umat muslim setelah Al-Qur'an. Hampir seluruh tata cara beribadah dalam islam dijelaskan dalam hadits secara mendetail. Penelitian ini dapat membantu umat muslim menemukan jenis kesahihan dari hadits yang beredar sekarang. Penulis mengklasifikasikan hadits menurut kesahihannya berdasarkan perawi hadits menggunakan metode *Backpropagation Neural Network* sebagai *classifier* dan *Principal Component Analysis* sebagai pereduksi dimensi fitur. Ada tiga target kategori yang digunakan dalam penelitian ini yaitu sahih, hasan, dan dhaif. Set data hadits sahih diambil dari kitab Sahih Bukhori, hasan dari kitab Sahih Sunan Tirmidzi, dan dhaif dari kitab Dhaif Abu Daud. Penulis memanfaatkan fungsi *split* dan *unique* dari bahasa pemrograman python untuk mengambil dan memfilter nama-nama perawi yang ada pada set data. Nama-nama perawi yang sudah terseleksi dikonversi menggunakan *Tf-Binary*. Setelah mencoba beberapa model pada proses validasi, didapatkan bahwa hasil akurasi yang terbaik adalah dengan menggunakan model PCA sebanyak 1500 fitur dari 3330 dan BPNN menggunakan satu *hidden layer* dengan jumlah *node* sebanyak 100 yakni sebesar 86,53%.

Kata Kunci— Klasifikasi Hadits, *Backpropagation Neural Network*, *Principal Component Analysis*, PCA, BPNN.

I. PENDAHULUAN

Hadits merupakan perkataan, perbuatan, ketetapan dari Rasulullah Muhammad SAW yang dijadikan sebagai sumber hukum kedua oleh umat muslim. Al-Hadits dirangkum melalui pewarisan Rosulullah SAW kepada para sahabatnya lalu sahabat itu ke satu yang lain dan begitu seterusnya -yang biasa dikenal sebagai sanad- sampai dapat diketahui oleh kita. Maka dari itu hadits digolongkan menjadi empat kategori berdasarkan kekuatan sanadnya yaitu, hadits sahih, hasan, dhaif, dan maudu'. Sahih yang berarti sangat kuat dan dijadikan landasan oleh banyak muslim, hasan yang bermakna kuat tapi masih adanya perawi yang kurang dalam hafalannya. Dhoif bermakna lemah yang disebabkan oleh beberapa faktor dari sifat perawinya. Dan terakhir adalah maudu' yakni hadits palsu yang dibuat-buat sendiri tetapi dinisbatkan kepada Rosulullah SAW.

Metode analisis kesahihan hadits ini dinilai lebih cepat dan lebih efisien jika dibandingkan cara konvensional yakni dengan mengecek suatu hadits dalam kitab-kitab hadits yang tebal. Cara ini dianggap lebih menguntungkan karena hanya perlu menyalin hadits yang akan dicek kesahihannya. Inilah sebabnya mengapa topik ini dapat dianggap mampu mengarahkan kepada cara yang lebih mudah dan cepat untuk menentukan tipe kesahihan hadits, dalam hal ini memanfaatkan nama-nama perawi hadits.

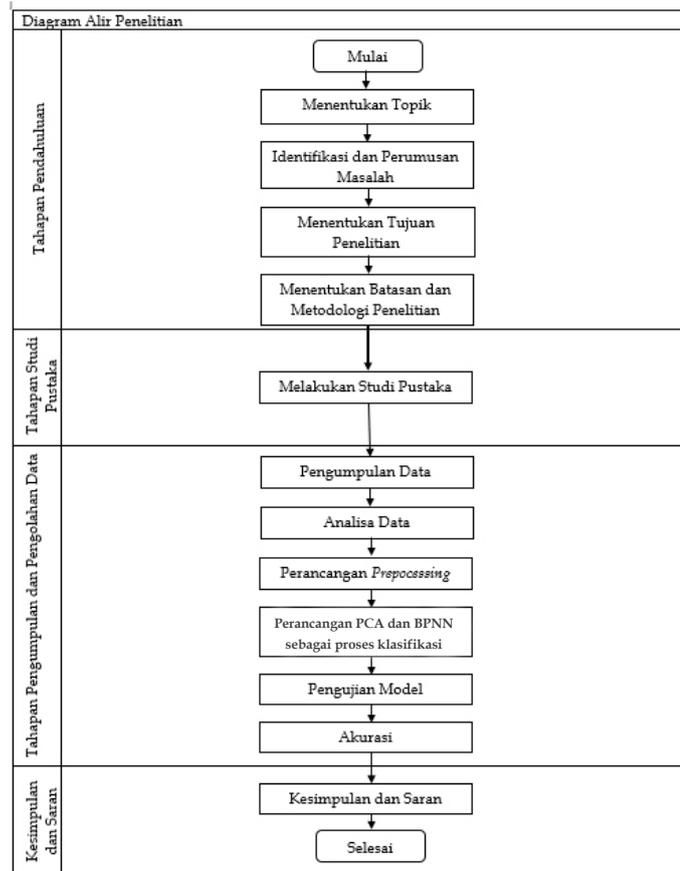
Banyak penelitian yang membahas tentang klasifikasi isi kandungan hadits tetapi tidak cukup banyak untuk topik ini. Salah satu penelitian yang penulis ambil sebagai bahan rujukan adalah jurnal internasional yang diteliti oleh Ghanem, Mouloudi, dan Mourchid dengan judul *Classification of Hadiths using LVQ based on VSM Considering Words Order*. Penelitian tersebut dijelaskan bahwa sebagai *classifier* penulis menggunakan algoritma *Vector Space Model*. Ghanem hanya menggunakan total 160 hadits sebagai datasetnya. Masing-masing kategori berjumlah 40 hadits dengan pembagian 75% digunakan sebagai data training dan sisanya 25% sebagai data testing. Setelah melalui proses *preprocessing* berupa penghapusan isi hadits, menghilangkan kata kerja, menghilangkan kata 'dari' dan standardisasi didapatkan ada 445 daftar nama perawi. Akurasi yang didapat adalah untuk sahih dan mudu' 80% dan 100% sedangkan hasan dan dhaif sebesar 20 % dan 0%.

Maksud dari pengambilan topik ini adalah untuk mengetahui tingkat kesahihan sebuah hadits dengan menganalisis nama-nama perawi yang ada pada hadits tersebut. Adapun tingkat kesahihan yang dimaksud terbatas pada tiga kategori yakni sahih, hasan, dan dhaif. Sedangkan untuk metode yang digunakan, penulis menggunakan *Principal Component Analysis* untuk mereduksi dimensi matriks yang digabungkan dengan *Backpropagation Neural Network* sebagai *classifier* dikarenakan dengan kedua metode ini dapat menghasilkan akurasi yang baik pada beberapa penelitian. Salah satunya pada penelitian Zhang, Zuo, Gao, dan Zhao dengan judul "*Digital Instruments Recognition Based on PCA-BP Neural Network Features*" dengan hasil akurasi sebesar 99% pada 100 citra dan 98.6% 1000 citra.

II. METODOLOGI PENELITIAN

Metode yang digunakan dalam melakukan penelitian ini adalah *Principal Component Analysis* dan *Backpropagation*

Neural Network. Terdapat tiga tahapan yang akan dilakukan pada penelitian ini, tahap yang pertama adalah tahap pendahuluan yang melakukan empat proses yang terdiri dari penentuan topik, identifikasi dan perumusan masalah, penentuan tujuan penelitian, dan penentuan batasan masalah serta metodologi penelitian. Pada tahap kedua yaitu tahapan studi pustaka. Bagian yang ketiga terdiri dari pengumpulan data, analisa data, rancangan preprocessing, perancangan PCA dan BPNN sebagai proses klasifikasi, dan pengujian model yang menghasilkan akurasi. Pada bagian akhir berisi kesimpulan dan saran dari penelitian yang telah dilakukan. Untuk memperjelas gambaran alur penelitian ini akan disajikan pada gambar 1 sebagai berikut :



Gbr. 1 Alur Proses Penelitian

A. Preprocessing Data

Preprocessing adalah tahapan sebelum teks diolah menggunakan arsitektur yang akan dibangun. Pada tahap preprocessing, akan dilakukan pencarian nama-nama perawi yang sebelumnya telah diapit dengan tanda kurung siku. Hal ini dilakukan untuk mengambil nama-nama perawi yang selanjutnya akan digunakan sebagai set data. Pengambilan nama-nama tersebut menggunakan fungsi *split* pada bahasa pemrograman python. Setelahnya dilakukan pembersihan nama perawi dari karakter-karakter yang tidak diperlukan baik berupa angka, *special character*, maupun *bad symbol*.

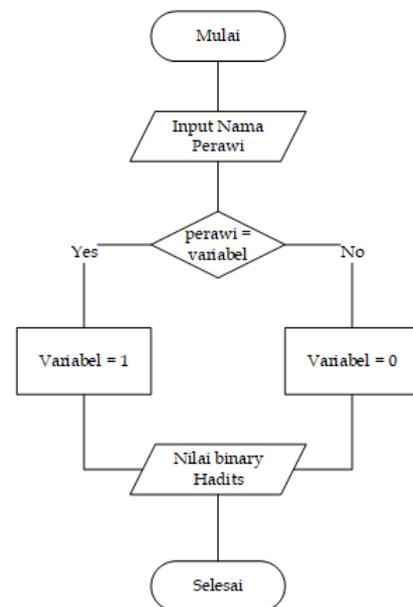
Pada Gambar 2 dibawah akan ditunjukkan alur dari *preprocessing*.



Gbr. 2 Alur Preprocessing

B. Konversi ke Tf-Binary

Setelah tahap preprocessing pada data dilakukan, maka selanjutnya akan dilakukan proses konversi ke *tf-binary*. Alur proses konversi ke *tf-binary* yang digunakan pada penelitian ini ditunjukkan pada Gambar 3:



Gbr. 3 Flowchart Tf Binary

Dari data perawi yang sudah terkumpul, keberadaan perawi yang ada dalam hadits diubah ke dalam angka binary,

yaitu jika ada diberi nilai 1 dan jika tidak ada diberi nilai 0. Konversi Tf-Binary ini baik digunakan pada topik ini karena kemungkinan munculnya satu nama perawi dalam satu hadits hanya satu kali, dengan kata lain tidak akan ada nama yang sama dalam satu hadits.

C. Proses Principal Component Analysis

Proses ini berfungsi untuk mengurangi dimensi data dengan mempertahankan sebanyak mungkin informasi dari dataset yang asli. Pada gambar 4 menunjukkan alur dari proses *Principal Component Analysis* yang digunakan dalam penelitian ini.



Gbr. 4 Proses Principal Component Analysis

1. Data yang diperoleh dari hasil ekstraksi fitur Tf Binary adalah data awal yang akan digunakan dalam proses Principal Component Analysis.
2. Mengurangi nilai Mean adalah tahap pertama yang dilakukan dengan cara mengurangi nilai rata-rata dari masing-masing dimensi data. Jadi semua nilai x dan y dari setiap data dikurangi. Pada akhirnya proses ini akan menghasilkan kumpulan data yang rata-ratanya nol.
3. Dalam memperoleh semua kemungkinan nilai kovarian dari dimensi yang berbeda-beda adalah dengan cara menghitungnya dengan matriks.

$$C^{Matrix} = (c_{i,j}, c_{ij} = cov(Dim_i, Dim_j)) \quad (1)$$

4. Proses selanjutnya adalah menghitung eigenvalue dan eigenvector dengan persamaan

$$Ax = \lambda x \quad (2)$$

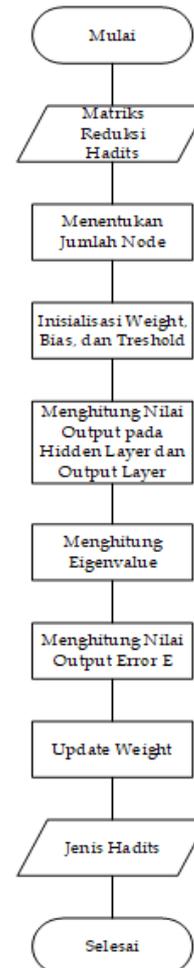
untuk memperoleh nilai eigen dan persamaan

$$det(A - \lambda I) = 0 \quad (3)$$

5. Adapun proses yang terakhir adalah dengan mengurangi dimensi sebesar K.

D. Proses Klasifikasi dengan Neural Network

Proses klasifikasi dilakukan ketika hadits telah diproses menggunakan PCA sehingga didapatkan hasil fitur yang telah tereduksi. Flowchart Algoritma *Neural Network* ditunjukkan pada Gambar 5.



Gbr. 5 Flowchart Algoritma Backpropagation Neural Network

1. Proses pertama menentukan jumlah node yang akan dipakai.
2. Inisialisasi weight, bias, dan threshold adalah dengan cara memberikan nilai pada masing-masing. Kemudian nilai weight yang didapatkan selanjutnya ditransform.
3. Menghitung nilai output pada hidden layer dapat dilakukan dengan rumus,

$$Y_j(p) = FA \left[\sum_{j=1}^n x_i \omega_{ij}(p) - \theta_j \right] \quad (4)$$

Kemudian *output layer* dengan rumus,

$$Y_k(p) = FA \left[\sum_{j=1}^n Y_j \omega_{jk}(p) - \theta_k \right] \quad (5)$$

4. Menghitung nilai error pada output layer adalah dengan perhitungan,

$$\delta_k(p) = Y_k(p) \times [1 - Y_k(p)] \times e_k(p) \quad (6)$$

dimana,

$$e_k(p) = target - aktivasi \quad (7)$$

5. Pada proses yang terakhir adalah memperbarui nilai bobot baik dari input layer ke hidden layer

$$\Delta \omega_{ij}(p) = \alpha \times \omega_{ij}(p) \times \delta_i(p) \quad (8)$$

$$\omega_{ij}(p+1) = \omega_{ij}(p) + \Delta \omega_{ij}(p) \quad (9)$$

maupun dari *hidden layer* ke *output layer*

$$\Delta \omega_{jk}(p) = \alpha \times Y_j(p) \times \delta_k(p) \quad (10)$$

$$\omega_{jk}(p+1) = \omega_{jk}(p) + \Delta \omega_{jk}(p) \quad (11)$$

E. Proses Validasi Model dengan K-Fold

Pada proses validasi bertujuan untuk mengevaluasi kinerja model yang ada. Pada proses ini dipisahkan dataset menjadi dua subset yang telah diolah. Dengan pembagian subset yang pertama sebagai subset pembelajaran dan yang kedua sebagai subset pengujian. Selain dapat mengurangi waktu komputasi, K-Fold Cross Validation juga dapat menjaga keakuratan estimasi. Pada penelitian ini menggunakan 10 fold karena cenderung memberikan estimasi akurasi yang akurat dibandingkan dengan yang biasa atau leave-one-out. Dalam 10 fold Cross Validation, data akan dibagi menjadi 10 dengan ukuran yang sama sehingga akan ada 10 subset data untuk mengevaluasi kinerja model atau algoritma. Untuk masing-masing dari 10 subset data tersebut, 9 fold digunakan untuk pelatihan dan 1 fold yang lain untuk pengujian.

F. Proses Perhitungan Akurasi

Proses perhitungan akurasi merupakan proses akhir dari penelitian ini. Setelah mendapatkan hasil dari proses pengujian, perhitungan akurasi dilakukan terhadap data testing. Persamaan untuk menghitung akurasi dapat dituliskan,

$$Akurasi = \frac{jumlah\ data\ benar}{total\ data} \times 100\% \quad (12)$$

III. HASIL DAN PEMBAHASAN

A. Analisis Hasil Validasi

Proses klasifikasi dengan BPNN (Backpropagation Neural Network) dilakukan di dalam proses validasi dengan 10 *Fold Cross Validation* dengan menentukan model untuk hidden layer dan node yang dipakai. Pada penelitian ini model yang dipakai mengadopsi dari teori deret konvergen atau deret yang berhingga. Dengan jumlah paling banyak 5 hidden layer dan paling sedikit 1 hidden layer dengan masing-masing layer memiliki jumlah node yang variatif yaitu 50, 100, 200, 300, 400, atau 500.

Model hasil yang terbaik menghasilkan rata-rata akurasi sebesar 86,5268% dengan model PCA sejumlah 1500 dan BPNN dengan satu hidden layer yakni 100 node. Lalu hasil terbaik tersebut disimpan dalam file berformat jlb yang selanjutnya digunakan sebagai data pada aplikasi yang disertai dengan *Graphical User Interface* (GUI).

1. Analisis Model Keseluruhan

Adapun hasil akurasi dari semua model yang dicoba telah dirangkum dalam sebuah tabel dan disortir secara *ascending* dari kecil ke besar menurut nilai *mean* dari seluruh fold. Terdapat enam model Principal Component Analysis dan sepuluh model Backpropagation Neural Network dengan jumlah total proses validasi ada 60 kali proses. Hasil rata-rata 10 fold cross validation yang terbaik dari semua model adalah dengan jumlah PCA 1500 dan BPNN dengan satu hidden layer yang mempunyai *nodes* sejumlah 100. Perolehan akurasi yang tinggi banyak terdapat pada fold 0. Dan didapatkan akurasi rata-rata hingga 90%. Berbeda dengan fold lainnya yang hanya mendapatkan rata-rata akurasi hanya 80% bahkan 70% saja.

2. Analisis Model berdasarkan Fold

Jumlah PCA yang ada pada Tabel 1 didominasi oleh jumlah PCA yang besar. Jika diambil nilai tengah dari 1000 sampai 1500 adalah 1250 maka ada tujuh fold yang mempunyai nilai di atasnya dengan rincian 1300 disebutkan satu kali, 1400 dua kali, dan 1500 empat kali. Dapat diambil kesimpulan bahwa semakin banyak jumlah PCA maka akurasi yang dihasilkan juga semakin besar.

Tabel 1 Akurasi Terbaik dari Setiap Fold

Fold	PCA	Hidden Layer (node)	Akurasi
0	1200	500-100	94,1489
1	1500	50	88,7701
2	1400	500-400-100	85,5615
3	1500	500-50	83,9572
4	1300	100	89,3048
5	1100	500-400-100	86,6310
6	1000	500-400-300-50	88,2353
7	1500	500-400-50	88,7701
8	1500	500-400-50	87,6344
9	1400	50	89,7849

Sementara pada jumlah hidden layer dan node dari Tabel 1 model yang memakai 1 (satu) hidden layer berjumlah 3 (tiga), yang memakai 2 (dua) hidden layer berjumlah 2 (dua), yang memakai 3 (tiga) hidden layer berjumlah 4 (empat), dan yang memakai 4 (empat) hidden layer hanya berjumlah 1 (satu). Sedangkan

untuk model yang menggunakan 5 (lima) hidden layer tidak ada. Dari data tersebut dapat diambil kesimpulan bahwa semakin banyak jumlah hidden layer maka akurasi yang dihasilkan semakin kecil dan sebaliknya.

B. Analisis Hasil Pengujian

Untuk mengetahui aplikasi yang dibuat sudah berjalan dengan benar maka dilakukan percobaan dengan memasukkan beberapa hadits dengan kategori yang berbeda ke dalam aplikasi web yang telah dibuat. Analisis yang dilakukan yaitu dengan memasukkan hadits-hadits baru sebanyak 60 hadits dengan pembagian masing-masing kategori berjumlah 20 hadits. Data yang diuji adalah selain data learning ataupun testing pada proses validasi. Hasil pengujian dirincikan pada Tabel 2, Tabel 3, dan Tabel 4. Tabel 2 merupakan hasil pengujian hadits sahih, Tabel 3 hadits hasan, dan Tabel 4 hadits dhaif. Hasil yang benar pada tabel-tabel tersebut ditandai dengan warna kuning.

Tabel 2 Hasil Pengujian Hadist Sahih pada Aplikasi

No	Nomor Hadits Sahih Kitab Sahih Muslim	Hasil
1	2	Shahih
2	3	Shahih
3	5	Hasan
4	6	Shahih
5	16	Shahih
6	17	Hasan
7	20	Shahih
8	22	Shahih
9	23	Shahih
10	24	Shahih
11	27	Hasan
12	29	Shahih
13	33	Shahih
14	36	Shahih
15	37	Shahih
16	39	Shahih
17	43	Shahih
18	45	Shahih
19	55	Shahih
20	56	Shahih

Tabel 3 Hasil Pengujian Hadist Hasan pada Aplikasi

No	Nomor Hadits Hasan Kitab Ibnu Majah	Hasil
1	18	Hasan
2	47	Hasan
3	116	Hasan

4	123	Dhaif
5	124	Hasan
6	140	Hasan
7	186	Hasan
8	230	Hasan
9	303	Hasan
10	346	Hasan
11	351	Hasan
12	352	Hasan
13	391	Hasan
14	392	Hasan
15	402	Shahih
16	407	Dhaif
17	445	Hasan
18	466	Hasan
19	471	Hasan
20	472	Shahih

Tabel 4 Hasil Pengujian Hadist Dhaif pada Aplikasi

No	Nomor Hadits Dhaif Kitab Sunan Darimi	Hasil
1	9	Dhaif
2	19	Dhaif
3	37	Dhaif
4	40	Sahih
5	48	Dhaif
6	50	Sahih
7	52	Sahih
8	57	Dhaif
9	60	Dhaif
10	68	Sahih
11	89	Dhaif
12	102	Sahih
13	108	Dhaif
14	109	Dhaif
15	114	Dhaif
16	115	Dhaif
17	119	Dhaif
18	125	Dhaif
19	129	Dhaif
20	148	Dhaif

Dari pengujian aplikasi yang telah dilakukan menghasilkan data yang benar sebanyak 48 dan salah berjumlah 12, sehingga jika menggunakan rumus perhitungan $Acc = \frac{\text{Data Benar}}{\text{Total Data}} \times 100\%$ akurasi yang didapat sebesar 80%.

IV. KESIMPULAN

Kesimpulan yang dapat diambil dari keseluruhan penelitian yang telah dilakukan yaitu penggunaan fitur teks hadits yang disertai dengan nama-nama perawi serta metode yang digunakan sebagai alat untuk mereduksi dimensi adalah Principal Component Analysis dan sebagai classifier menggunakan Backpropagation Neural Network. Hasil keluaran dari penelitian ini berupa tiga kategori hadits Sahih, Hasan, dan Dhaif. Dengan mencoba beberapa model didapati bahwa model yang terbaik adalah dengan menggunakan reduksi PCA sampai 1500 fitur dan hidden layer yang digunakan dalam proses Backpropagation Neural Network sebanyak satu dengan jumlah node sebanyak 100. Dengan model tersebut proses klasifikasi dapat menghasilkan akurasi sebesar 86,5689 %.

Berdasarkan analisis yang telah dilakukan penulis menyarankan agar mencoba untuk mengembangkan dengan menambah jumlah data yang digunakan mengingat jumlah hadits keseluruhan mencapai puluhan ribu, dapat diaplikasikan dalam Bahasa Arab karena hadits yang sebenarnya ditulis menggunakan Bahasa Arab, menambah jumlah kategori lainnya seperti maudu' (palsu), dan mencoba untuk menggunakan model ataupun metode lainnya yang dapat menghasilkan akurasi yang lebih besar sebagai contoh algoritma *Vector Space Model* (VSM).

UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya kepada Allah SWT yang telah memberikan pertolongannya dalam setiap langkah pengerjaan penelitian ini. Terimakasih pula untuk semua pihak yang telah memberikan dukungan sehingga penelitian ini dapat berjalan dan terselesaikan dengan baik.

REFERENSI

- [1] Abdulhussain, M. I., & Gan, J. Q. (2015). An experimental investigation on PCA based on cosine similarity and correlation for text feature dimensionality reduction. *IEEE*.
- [2] Al-Asqalant, I. H. (n.d.). *Ilmu Mustalah Hadits Dasar*. Panjiaswaja Pres.
- [3] Al-Maliki, S. M. (2000). *Al-Manhalul Lathif fi Ushulil Hadits As-Syarif*. Madinah: Maktabah Malik Fahd.
- [4] Al-Qahthan, M. (2009). *Pengantar Studi Ilmu Hadits*. Maktabah Wahbah.
- [5] Bakar, A., Adiwijaya, & Faraby, S. (2018). Multi-label topic classification of Hadith Bukhari (Indonesian Language Translation) using Information Gain and Backpropagation Neural Network. Bandung.
- [6] Ghanem, M., Mouloudi, A., & Murchid, M. (2016). Classification of Hadiths using LVQ based on VSM

- Considering Words Order. *International Journal of Computer Applications* (0975 – 8887), 148-No. 4.
- [7] Idri, D. (n.d.). *Studi Hadits*. Kencana.
- [8] JUD. (2016). *Pemrograman Python untuk Pemula* (First ed.). Yogyakarta: CV Jubilee Solusi Enterprise.
- [9] Liu, L., & Ozsu, M. T. (2009). *Encyclopedia of Database System*. Research Gate.
- [10] Zhang, J., Zuo, L., Gao, J., & Zhao, S. (2017). Digital instruments recognition based on PCA-BP neural network.