



Implementasi Algoritma Winnowing Pada Aplikasi Kisah Para Nabi dan Rasul

Mirawati

Program Studi Teknik Informatika, Fakultas Teknik Informatika, Universtias Budi Darma Medan, Indonesia

Jl. Sisingamangaraja No.338, Siti Rejo I, Kec. Medan Kota, Kota Medan, Sumatera Utara, Indonesia

Email: mirawati2512@gmail.com

Abstrak—Buku merupakan media pembelajaran umum yang digunakan dalam menceritakan kisah nabi, namun media seperti buku memiliki beberapa kekurangan fisik yang mudah rusak, dan kurang interaktif yang menyebabkan mudah bosan sehingga diperlukan sebuah aplikasi pembelajaran tentang kisah para nabi dan rasul yang dapat meningkatkan minat membaca tentang kisah para nabi dan rasul pada masyarakat. Adapun tujuan dilakukan penelitian ini untuk mengetahui prosedur aplikasi para nabi dan rasul. Menerapkan aplikasi yang mengimplementasikan algoritma winnowing pada aplikasi kisah para nabi dan rasul. Adapun manfaat yang didapat dengan menerapkan algoritma winnowing pada aplikasi kisah para nabi dan rasul adalah untuk meningkatkan minat membaca tentang kisah para nabi dan rasul.

Kata Kunci: Algoritma Winnowing, Aplikasi, Kisah, Nabi dan Rasul

Abstract—Books are a common learning medium used in telling the story of the prophet, but media such as books have several physical deficiencies that are easily damaged, and are less interactive which cause boredom easily, so we need a learning application about the stories of the Prophets and Messengers that can increase reading interest in the stories of the Prophets and Messengers in society. The purpose of this research is to determine the application procedure of the Prophets and Messengers. Implement an application that implements the winnowing algorithm in the application of the stories of the Prophets and Messengers. The benefits obtained by applying the winnowing algorithm in the application of the stories of the Prophets and Messengers are to increase interest in reading about the stories of the Prophets and Messengers.

Keywords: Winnowing Algorithm, Application, Story, Prophets and Messengers

1. PENDAHULUAN

Tidak sedikit masyarakat didaerah yang menganut agama Islam yang belum memahami lebih dalam tentang agama Islam. Oleh karena itu pengetahuan agama Islam sangat penting dan wajib dipelajari oleh setiap muslim yang perlu ditanamkan sejak dini. Salah satu kewajiban umat muslim adalah iman kepada Nabi dan Rasul-Rasul Allah SWT dengan cara mengetahui dan meyakini adanya Nabi dan Rasul-Rasul Allah SWT serta mencontoh akhlak baik yang dimiliki oleh para Nabi dan Rasul karena iman kepada Nabi dan Rasul merupakan salah satu dari rukun iman. Hal ini termaktub dalam Q.S An Nisaa ayat 136 yang artinya, “Wahai orang-orang yang beriman, tetapkanlah beriman kepada Allah dan Rasul-Nya...” (Saifuddin A, 2004)

Kisah Nabi dan Rasul dalam Islam merupakan kisah yang dapat dijadikan pembelajaran dalam mencontoh akhlak para Nabi dan Rasul. Setelah mempelajari Kisah Nabi dan Rasul, evaluasi juga merupakan bagian yang sangat penting karena evaluasi bertujuan untuk memperbaiki proses pembelajaran kisah Nabi dan Rasul dengan mengetahui bagian mana yang belum dipahami oleh pembaca. Pada umumnya kisah Nabi dan Rasul diceritakan ataupun dibaca karena sifat dan akhlak para Nabi dan Rasul yang sangat baik untuk dicontoh dan ditiru dan juga dapat berkembang menjadi pribadi yang memiliki akhlak yang baik.

Dalam hal pemahaman tersebut, diperlukan sebuah aplikasi untuk membantu dalam mempermudah pencarian cerita ataupun kisah-kisah para Nabi dan Rasul tersebut. Dimana pada aplikasi tersebut dibutuhkan suatu teknik yang disebut dengan *String Matching*. *String Matching* merupakan sebuah teknik untuk melakukan pencocokan pada *string* ataupun kata dimana dipergunakan terhadap permasalahan untuk mencari pola susunan kata ataupun karakter (Br Ginting & Utomo, 2019). Ada banyak algoritma yang dapat digunakan pada *string matching* tersebut salah satunya adalah algoritma *winnowing*.

Algoritma *winnowing* merupakan salah satu metode kemiripan teks (*text similarity*) yang merupakan bagian dari *text mining*. Digunakannya metode kemiripan teks (*text similarity*) dikarenakan metode ini dapat memenuhi kebutuhan informasi tingkat kemiripan dokumen secara cepat dan akurat. maka dalam prosesnya menggunakan algoritma *winnowing* sebagai pengukur tingkat kemiripan dokumen dengan kata kunci yang didapat hasil ekstraksi teks pada dokumen. Algoritma *winnowing* melakukan penghitungan nilai-nilai *hash* dari setiap *k-gram*, untuk mencari nilai *hash* selanjutnya digunakan fungsi *rolling hash*. Kemudian dibentuk *window* dari nilai-nilai *hash* tersebut. Dalam setiap *window* dipilih nilai *hash* minimum. Jika ada lebih dari satu *hash* dengan nilai minimum, dipilih nilai *hash* yang paling kanan. Kemudian semua nilai *hash* terpilih disimpan untuk dijadikan *fingerpint* dari suatu dokumen. *Fingerpint* ini yang akan dijadikan dasar pembandingan kesamaan antara teks yang telah dimasukkan.

Penerapan algoritma *winnowing* pada Aplikasi Kisah Para Nabi dan Rasul bertujuan untuk dapat mempermudah pencarian kisah Para Nabi dan Rasul pada masyarakat pada umumnya. Maka dengan diterapkannya algoritma *winnowing* dalam Aplikasi Kisah Para Nabi dan Rasul kedalam sebuah Aplikasi yang dapat dijadikan teladan terutama dala mencontoh akhlak para Nabi dan Rasul. Pada penelitian terdahulu yang dilakukan oleh Jarwati dkk, yang berjudul “Penerapan Algoritma Winnowing pada Sistem Rekomendasi Penentuan Dosen Pembimbing Skripsi (Studi Kasus



Program Studi Sistem Informasi)” mengatakan bahwa penerapan algoritma winnowing dapat memberikan perhitungan kesamaan kata kunci terhadap apa yang dicari (Jarwati, Prihandoko, Cahya, & R, 2017).

2. METODOLOGI PENELITIAN

2.1 Metodologi Penelitian

Dalam melakukan penelitian ini, metodologi penelitian yang dipakai adalah sebagai berikut.

1. Pengumpulan Data
Kegiatan ini bertujuan untuk mengumpulkan data atau informasi yang dibutuhkan dari buku-buku dan jurnal-jurnal yang ada di perpustakaan atau internet yang berhubungan dengan judul penelitian.
2. Tahap Analisa
Pada tahap ini, perencanaan aplikasi akan dibangun berdasarkan permasalahan yang sedang dihadapi dengan menganalisa proses pencarian kisah para Nabi dan Rasul dengan menggunakan algoritma Winnowing.
3. Tahap Perancangan
Tahap ini merupakan tahap bagi para pengembang untuk melakukan desain secara lengkap berdasarkan hasil analisa.
4. Tahap Pengujian dan Implementasi
Pada tahap ini pemeriksaan kembali dari tahap sebelumnya kemudian diimplementasikan hasil desain dan melakukan pengujian dari implementasi.
5. Dokumentasi
Dokumentasi merupakan tahapan untuk penyusunan laporan dari hasil penelitian yang telah dilakukan.

2.2 String Matching

String Matching adalah proses pencarian semua kemunculan *query* yang selanjutnya disebut *pattern* ke dalam *string* yang lebih panjang (teks). *Pattern* dilambangkan dengan $x=x[0..m-1]$ dan panjangnya adalah m . Teks dilambangkan dengan $y=y[0..n-1]$ dan panjangnya adalah n . Kedua *string* terdiri dari sekumpulan karakter yang disebut alfabet yang dilambangkan dengan Σ dan mempunyai ukuran σ . *String matching* dibagi menjadi dua, yakni *exact matching* dan *heuristic* atau *statistical matching*. Algoritma *string matching* adalah suatu metode yang digunakan untuk menemukan suatu keakuratan atau hasil dari satu atau beberapa pola teks yang diberikan. *String matching* merupakan pokok bahasan yang penting dalam ilmu komputer karena teks merupakan merupakan bentuk utama dari pertukaran informasi antar manusia, misalnya pada literatur, karya ilmiah, halaman web, dan sebagainya. Dalam algoritma *string matching*, teks diasumsikan berada di dalam *memory*, sehingga bila mencari *string* di dalam sebuah teks, maka semua isi teks perlu dibaca terlebih dahulu kemudian disimpan didalam *memory*. *String matching* fokus pada pencarian satu, atau lebih umum, semua kehadiran sebuah kata (*pattern*) dalam sebuah teks (Br Ginting & Utomo, 2019).

- a. *Exact Matching*, digunakan untuk menemukan *pattern* yang berasal dari satu teks. Contoh pencarian *exact matching* adalah pencarian kata “pelajar” dalam kalimat “saya seorang pelajar” atau “saya seorang siswa”. Sistem akan memberikan hasil bahwa kalimat pertama mengandung kata “pelajar” sedangkan kalimat kedua tidak, meskipun kenyataannya pelajar dan siswa adalah kata yang bersinonim.
- b. *Heuristic matching* adalah teknik yang digunakan untuk menghubungkan dua data terpisah ketika *exact matching* tidak mampu mengatasi karena ada pembatasan pada data yang tersedia. *Heuristic matching* dapat dilakukan dengan perhitungan *distance* antara *pattern* dengan teks.

N-gram similarity sering digunakan untuk membandingkan beberapa *string*. Metode ini menghitung sejumlah n - v gram bersama seperti serangkaian di antara *string*. Sebagai contoh bigram untuk *string article* adalah “ar”, “rt”, “ti”, “ic”, “cl”, “le”, sedangkan untuk trigram adalah “art”, “rti”, “tic”, “icl”, “cle”. Perhitungan kemiripan dua *string* dapat diukur menggunakan *jaccard coefficient* yaitu

$$\text{Jaccard Coefficient}(\text{pattern}, \text{text}) = \frac{|\text{pattern} \cap \text{text}|}{|\text{pattern} \cup \text{text}|}$$

2.3 Algoritma Winnowing

Algoritma winnowing adalah salah satu algoritma pencocokan string. Pada pencocokannya. Algoritma winnowing harus memenuhi kebutuhan mendasar yaitu (Sutrisno, 2005):

- a. *Whitespace insensitivity* yaitu pencarian kalimat mirip seharusnya tidak terpengaruh oleh spasi, jenis huruf (kapital atau normal), tanda baca dan sebagainya.
- b. *Noise suppression* yaitu menghindari penemuan kecocokan dengan panjang kata yang terlalu kecil atau kurang relevan seperti “the” dan bukan merupakan kata yang umum digunakan.
- c. *Position independence* yaitu penemuan kesamaan harus tidak bergantung pada posisi kata-kata sehingga kata dengan urutan posisi berbeda masih dapat dikenali jika terjadi kesamaan.
 1. Pembuangan karakter yang tidak relevan



Pembuangan karakter yang tidak relevan memenuhi kebutuhan algoritma winnowing yaitu *whitespace insensitivity*. Pada tahap ini proses yang dilakukan yaitu penghapusan tanda baca, spasi dan simbol-simbol seperti @, #, \$, %, *, (,), !, -, _ , " , + , > , < / dan sebagainya.

2. Pembentukan Rangkaian *n_gram* pada algo- ritma *winnowing* dilakukan dengan cara mem- bentuk rangkaian karakter sepanjang *n* dari hasil pembuangan karakter yang tidak relevan. Nilai *n* yang baik tidak terlalu kecil dan juga tidak terlalu besar. Rangkaian *n-gram* pertama dimulai dari karakter ke-1 sampai karakter ke-*n* dan rangkaian kedua dimulai dari karakter k-2 sampai ke- *n*+1 dan seterusnya sampai terbentuk rangkaian *n-gram* semua karakter.
3. Perhitungan Fungsi *Hash* untuk setiap *n_gram*

Algoritma *winnowing* menggunakan *rolling hash* untuk menghitung nilai Hash masing-masing rangkaian *gram*. Fungsi *hash* dengan *rolling hash* didefinisikan pada persamaan (1).

$$H = c1 * bk - 1 + c2 * bk - 2 + c3 * bk - 3 + ... + ck * b0 \dots\dots\dots (1)$$

dengan *c* adalah nilai *ascii* suatu karakter, *b* atau basis merupakan bilangan prima (tidak ditentukan) dan *n* adalah banyaknya karakter atau panjang rangkaian *n-gram*. Untuk nilai *hash* kedua dan selanjutnya, perhitungan tidak perlu melakukan iterasi dari indeks pertama sampai terakhir. Perhitungan nilai hash dapat dilakukan dengan cara:

$$H = c1 * bk - 1 + c2 * bk - 2 + c3 * bk - 3 + ... + ck * b0$$

4. Pembentukan *Window* dari nilai *Hash*
Algoritma *winnowing* tidak menggunakan semua nilai *hash* dari setiap rangkaian *gram* yang dibentuk. Nilai *hash* yang dibentuk pada tahap sebelumnya akan dibagi ke dalam *window* berukuran *w*. *Window* pertama berisi nilai *hash* pertama sampai nilai *hash* ke-*w*. *Window* kedua dibentuk dari nilai *hash* kedua sampai nilai *hash* ke-*w*+1 dan seterusnya sampai terbentuk *window* dari seluruh nilai *hash*.
5. Pemilihan *Fingerprint* dari setiap *Window* Nilai *fingerprint* yang dibentuk dari algoritma *winnowing* digunakan untuk mengukur prosentase kemiripan teks pada persamaan (3), persamaan *Jaccard Coefficient*.

$$Similarity = \frac{Jumlah\ fingerprint\ sama}{Total\ seluruh\ fingerprint} \times 100\% \dots\dots\dots (2)$$

atau...

$$Similarity (di, di) = \frac{|w (di) \cap w (dj)|}{|w (di) \cup w (di)|} \times 100\% \dots\dots\dots (3)$$

Dengan *d_i* nilai-nilai *fingerprint* pada teks, *d* nilai-nilai *fingerprint* pada teks, *w (di) ∩ w (dj)* jumlah nilai *fingerprint* sama antara teks ke-*i* dan teks ke-*j* dan *w (di) ∪ w (di)* adalah total nilai *fingerprint* teks ke-*i* dan teks ke *j*.

2.5 Kisah Nabi dan Rasul

Nabi berasal dari kata naba, artinya dari tempat yang tinggi. Sementara nabi secara umum Nabi artinya manusia (lelaki) seorang hamba Allah SWT yang diberi wahyu oleh Allah SWT (Martiani, 2013).

Kisah Nabi dan Rasul dalam Islam merupakan kisah yang dapat dijadikan pembelajaran dalam mencontoh akhlak para Nabi dan Rasul, setelah mempelajari kisah Nabi dan Rasul evaluasi juga merupakan bagian yang sangat penting karena evaluasi bertujuan untuk memperbaiki proses pembelajaran kisah Nabi dan Rasul dengan mengetahui bagian mana yang belum dipahami oleh pembaca. Dalam ajaran Islam kita wajib mengetahui dan mengenal 25 Nabi dan Rasul yang terdapat di dalam Al-Qur'an mulai dari Nabi Adam AS sampai dengan Nabi Muhammad SAW. Pada umumnya kisah Nabi dan Rasul diceritakan ataupun dibaca karena sifat dan akhlak para Nabi dan Rasul yang sangat baik untuk dicontoh dan ditiru dan juga ditiru serta dapat berkembang menjadi pribadi yang memiliki akhlak yang baik (Khoitunnisa, 2012).

3. HASIL DAN PEMBAHASAN

3.1 Analisa Algoritma Winnowing

Untuk melakukan proses pada algoritma winnowing dengan mencocokkan nilai Hash, terlebih dahulu diketahui nilai ASCII yang menjadi *Text* dan juga *Pattern* pada pencocokan string yang akan dicari. Adapun kode *Text* yang akan digunakan adalah "MUHAMMAD SAW", sehingga dapat dilihat kode ASCII pada tabel di bawah ini.

Tabel 1. Nilai ASCII

No	Char	Kode ASCII
1	M	077
2	U	085
3	H	072
4	A	065
5	D	068



Pada kasus ini, penulis akan membuat contoh dari Kisah para Nabi dan Rasul, misalnya dengan teks Nabi MUHAMMAD SAW yang menggunakan kata "MUHAMMAD" sebagai *pattern*-nya maka proses akan berbentuk sebagai berikut.

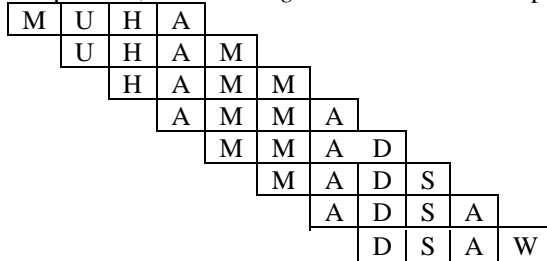
Contoh Kasus:

Teks : MUHAMMAD SAW

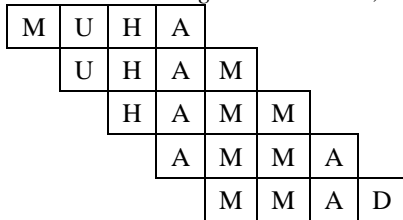
Pattern : MUHAMMAD

a. Parsing *n_gram*

Untuk pertama, membuat *n_gram* n=4 untuk teks seperti dibawah ini.



Setelah membuat *n_gram* untuk teks, selanjutnya membuat *n_gram* n=4 untuk pattern seperti di bawah ini.



b. Perhitungan nilai Hash pada *n_gram*.

Perhitungan nilai hash pada rangkaian *n_gram* dengan nilai basis (b) = 2, panjang rangkaian *n_gram*(n) = 4 sehingga dapat dihitung untuk nilai Hash setiap *n_gram* dan juga Pattern dengan menggunakan persamaan sebagai berikut:

$$H = \text{asci}(b) * 2^3 + \text{asci}(e) * 2^2 + \text{asci}(l) * 2^1 + \text{asci}(a)$$

1. Nilai Hash *n_gram* Teks

a. *n_gram* 1 [MUHA]

$$\begin{aligned}
H &= 77 * 2^{4-1} + 85 * 2^{4-2} + 72 * 2^{4-3} + 65 * 2^{4-4} \\
&= 77 * 2^3 + 85 * 2^2 + 72 * 2^1 + 65 * 2^0 \\
&= 77 * 8 + 85 * 4 + 72 * 2 + 65 * 1 \\
&= 616 + 340 + 144 + 65 \\
&= 1165
\end{aligned}$$

b. *n_gram* 2 [UHAM]

$$\begin{aligned}
H &= 85 * 2^{4-1} + 72 * 2^{4-2} + 65 * 2^{4-3} + 77 * 2^{4-4} \\
&= 85 * 2^3 + 72 * 2^2 + 65 * 2^1 + 77 * 2^0 \\
&= 85 * 8 + 72 * 4 + 65 * 2 + 77 * 1 \\
&= 680 + 288 + 130 + 77 \\
&= 1175
\end{aligned}$$

c. *n_gram* 3 [HAMM]

$$\begin{aligned}
H &= 72 * 2^{4-1} + 65 * 2^{4-2} + 77 * 2^{4-3} + 77 * 2^{4-4} \\
&= 72 * 2^3 + 65 * 2^2 + 77 * 2^1 + 77 * 2^0 \\
&= 72 * 8 + 65 * 4 + 77 * 2 + 77 * 1 \\
&= 576 + 260 + 154 + 77 \\
&= 1067
\end{aligned}$$

Dan langkah selanjutnya AMMA, MMAD, MADS, ADSA, DSAW. Setelah dilakukan perhitungan nilai Hash, maka didapatkan nilai Hash *n_gram* Teks 1 sampai dengan *k_gram* Teks 8 dengan nilai Hash [1165, 1175, 1067, 1047, 1122, 1095, 1023, 1093].

2. Nilai Hash *n_gram* Pattern

a. *n_gram* 1 [MUHA]

$$\begin{aligned}
H &= 77 * 2^{4-1} + 85 * 2^{4-2} + 72 * 2^{4-3} + 65 * 2^{4-4} \\
&= 77 * 2^3 + 85 * 2^2 + 72 * 2^1 + 65 * 2^0 \\
&= 77 * 8 + 85 * 4 + 72 * 2 + 65 * 1 \\
&= 616 + 340 + 144 + 65 \\
&= 1165
\end{aligned}$$



b. *n-gram* 2 [UHAM]

$$\begin{aligned}
H &= 85 * 2^{4-1} + 72 * 2^{4-2} + 65 * 2^{4-3} + 77 * 2^{4-4} \\
&= 85 * 2^3 + 72 * 2^2 + 65 * 2^1 + 77 * 2^0 \\
&= 85 * 8 + 72 * 4 + 65 * 2 + 77 * 1 \\
&= 680 + 288 + 130 + 77 \\
&= 1175
\end{aligned}$$

c. *n-gram* 3 [HAMM]

$$\begin{aligned}
H &= 72 * 2^{4-1} + 65 * 2^{4-2} + 77 * 2^{4-3} + 77 * 2^{4-4} \\
&= 72 * 2^3 + 65 * 2^2 + 77 * 2^1 + 77 * 2^0 \\
&= 72 * 8 + 65 * 4 + 77 * 2 + 77 * 1 \\
&= 576 + 260 + 154 + 77 \\
&= 1067
\end{aligned}$$

Dan langkah selanjutnya AMMA, MMAD. Setelah dilakukan perhitungan nilai *Hash*, maka didapatkan nilai *Hash n-gram Pattern* 1 sampai dengan *n-gram Pattern* 5 dengan nilai *Hash* [1165, 1175, 1067, 1047, 1122].

3. Pembentukan *window*

Pembentukan *window* dari hasil perhitungan nilai *hash teks* pada tahap sebelumnya dengan ukuran lebar *window* (w) = 3, *window* MUHAMMAD SAW yaitu.

- [1165 1175 1067]
- [1175 1067 1047]
- [1067 1047 1122]
- [1047 1122 1095]
- [1122 1095 1023]
- [1095 1023 1093]

Pembentukan *window* dari hasil perhitungan nilai *hash pattern* pada tahap sebelumnya dengan ukuran lebar *window* (w) = 3, *window* MUHAMMAD SAW yaitu.

- [1165 1175 1067]
- [1175 1067 1047]
- [1067 1047 1122]

4. Pemilihan nilai *fingerprint*

Pemilihan nilai *fingerprint* dari hasil pembentukan *window* pada tahap sebelumnya, *fingerprint teks* yaitu.

- [1165 1175 1067] = 1067
- [1175 1067 1047] = 1047
- [1067 1047 1122] = 1047
- [1047 1122 1095] = 1047
- [1122 1095 1023] = 1023
- [1095 1023 1093] = 1023

Pemilihan nilai *fingerprint* dari hasil pembentukan *window* pada tahap sebelumnya, *fingerprint pattern* yaitu.

- [1165 1175 1067] = 1067
- [1175 1067 1047] = 1047
- [1067 1047 1122] = 1047

Fingerprint yang terbentuk yaitu.

Fingerprint Teks = 1023, 1047, 1067

Fingerprint Pattern = 1047, 1067

5. Perhitungan kesamaan

Perhitungan kesamaan dengan menggunakan persamaan jaccard coefficient yaitu.

Fingerprint Teks = 1023, 1047, 1067

Fingerprint Pattern = 1047, 1067

Jumlah *fingerprint* yang sama (1023, 1047, 1067) = 3

Keseluruhan *fingerprint* =(1047, 1067) = 2

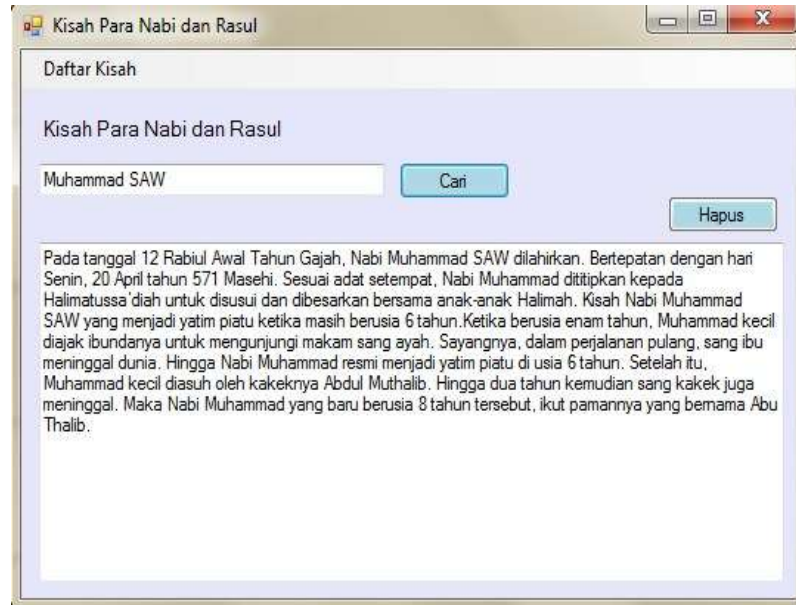
Proses selanjutnya menghitung *similitary* yaitu tingkat kesamaan antara *text* dan *pattern*, yaitu berapa persen tingkat kesamaannya. Berikut ini adalah proses menghitung *similitary* dua dokumen di atas:

$$\begin{aligned}
\text{Similitary (Text,Pattern)} &= \frac{\sum H_{\text{text}} \cap \sum H_{\text{pattern}}}{\sum H_{\text{text}} \cup \sum H_{\text{pattern}}} \times 100\% \\
&= \frac{2}{2+3-2} * 100\% \\
&= \frac{2}{3-2} * 100\% \\
&= \frac{2}{3} * 100\% \\
&= 67\%
\end{aligned}$$

3.2 Hasil Implementasi dan Pengujian

3.2.1 Tampilan Menu Utama

Tampilan menu utama merupakan tampilan awal dari aplikasi kisah para Nabi dan Rasul berfungsi untuk mencari kisah Nabi. Untuk lebih jelas merujuk pada gambar dibawah.



Gambar 1. Tampilan Menu Utama

3.2.2 Tampilan Menu Daftar Kisah

Tampilan menu daftar kisah berfungsi untuk menampilkan daftar Kisah Nabi dan Rasul. Untuk lebih jelas merujuk pada gambar dibawah.



Gambar 2. Tampilan Menu Daftar Kisah

3.3 Perancangan Aplikasi

Perancangan pencarian pada aplikasi Kisah Para Nabi dan Rasul berbasis Dekstop dirancang dengan menggunakan *Microsoft Visual Basic 2008*. *Microsoft Visual Basic 2008* adalah editor yang digunakan untuk pengguna menuliskan kode program dan mendesain tampilan aplikasi. Adapun pilihan menu pada tampilan aplikasi yang akan dibuat penulis yaitu berupa menu pencarian nabi dan rasul.

3.4 Algoritma

Algoritma merupakan prosedur langkah-langkah dalam penyelesaian masalah yang disusun secara sistematis dan logis. Pada umumnya algoritma dapat diuraikan dalam 3 bentuk deskripsi yaitu *Flowchart*, *Pseudocode* dan Deskripsi uraian untuk menjelaskan kerangka awal (ide) suatu program dengan jelas. Berikut penulisan *Pseudocode* pada aplikasi Kisah Para Nabi dan Rasul menggunakan algoritma winnowing.

Function Winnowing (input *s*: String [1..m], teks : string [1..n])

Boolean

Deklarasi

i = Integer

ketemu = Boolean

Algoritma

ketemu ← False

hs ← hash (s[1..m])

For *i* ← 0 to n-m Do

 hsub ← hash (teks[1..i+m-1])

 If hsub = hs Then

 If teks [i..i+m-1] = s Then

ketemu ← True

 Else

 hsub ← hash (teks[i+1..i+m])

Endfor

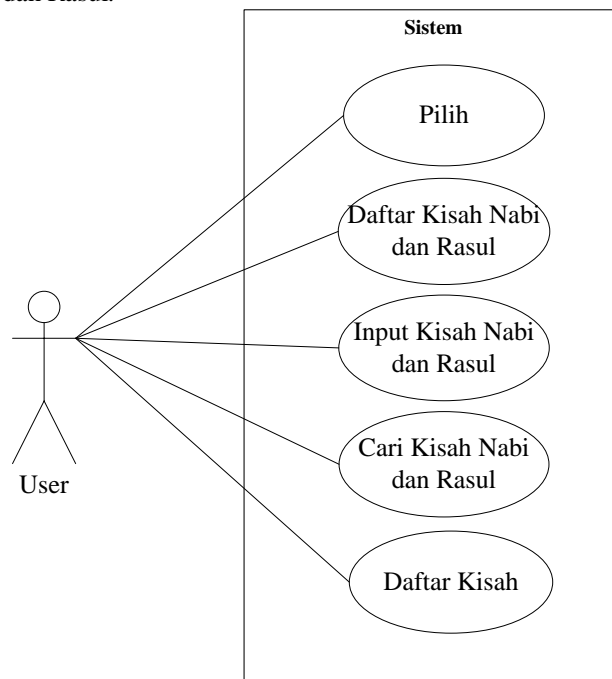
Return ketemu

3.5 Pemodelan

Penulis melakukan perancangan aplikasi Kisah Para Nabi dan Rasul berbasis *desktop* dengan pemodelan yang dibangun dengan menggunakan UML (*Unified Modeling Language*) dan perancangan *Interface*.

3.5.1 Use Case Diagram

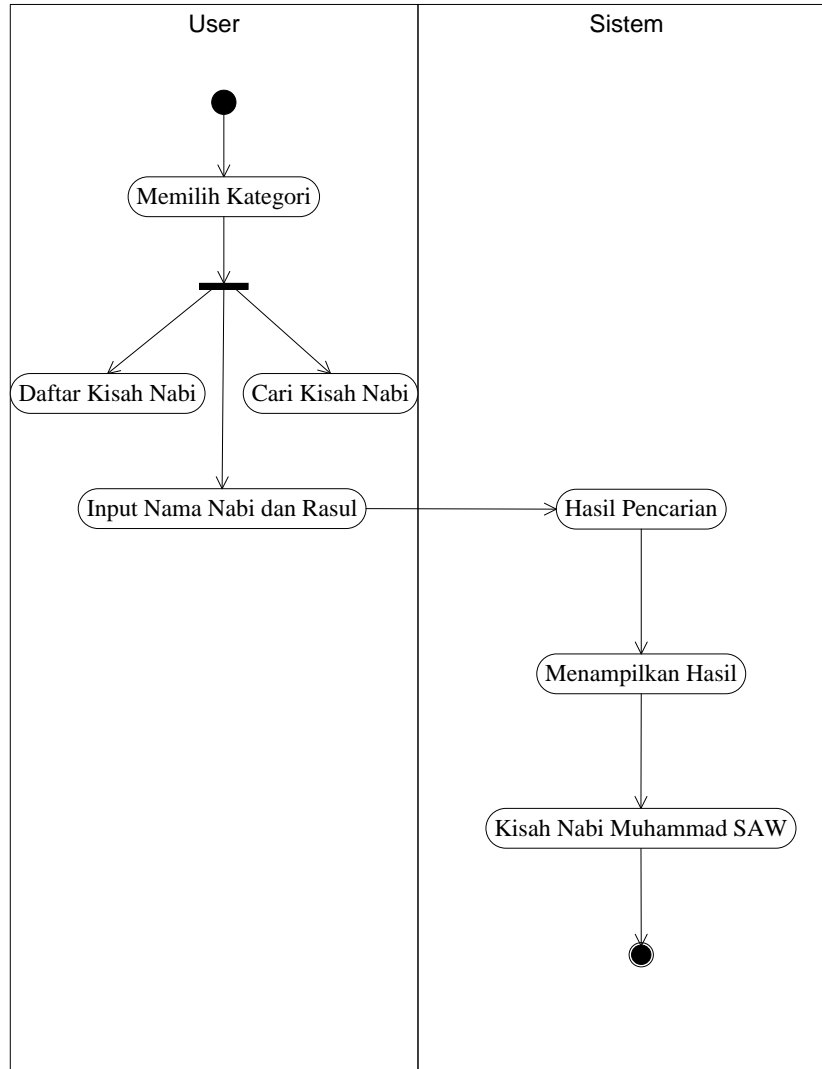
Use case diagram ini menjelaskan dan menerangkan kebutuhan yang digunakan oleh pengguna (*user*), dan sangat berguna dalam menentukan struktur organisasi dan model dari sebuah sistem. Berikut adalah gambar *use case diagram* pada aplikasi Kisah Para Nabi dan Rasul.



Gambar 3. Use Case Diagram

3.5.2 Activity Diagram

Activity diagram juga digunakan untuk mendefinisikan urutan atau pengelompokan dari sistem atau *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan serta rancangan menu yang ditampilkan pada Kisah Para Nabi dan Rasul.



Gambar 4. Activity Diagram

Dari gambar di atas menunjukkan aktivitas diagram pencarian, pertama user dapat menginput kata, kemudian system akan membuka database, setelah database dibuka akan menampilkan kata dan menampilkan daftar saran kata yang dicari oleh pengguna sesuai huruf yang diketikan. Kemudian user dapat memilih kata yang ada dalam daftar yang di cari. Jika kata yang dicari tidak ada, maka kata tersebut tidak ada dalam database.

3.6 Database

Penulis merancang *Database* aplikasi Kisah para Nabi dan Rasul dengan menggunakan *software*. Rancangan tabel *Database* aplikasi Kisah para Nabi dan Rasu, sebagai berikut:

1. Nama Database : Nabi_Rasul
Nama Tabel : kisah

Tabel 2. Tabel Database Kisah

<i>Field</i>	<i>Type</i>	<i>Size</i>
nama	varchar	25
kisah	mediumtext	

2. Nama Database : Nabi_Rasul
Nama Tabel : daftar

Tabel 3. Tabel Database Daftar

<i>Field</i>	<i>Type</i>	<i>Size</i>
no	integer	3
daftar	varchar	25



4. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan oleh penulis tentang aplikasi kisah para Nabi dan Rasul, maka dari bab-bab sebelumnya dapat diambil kesimpulan yang merupakan hasil akhir dari penulisan yaitu Proses pencarian kisah para Nabi dan Rasul dapat dilakukan dengan meng-input-kan kata yang ingin dicari. Kemudian aplikasi akan menampilkan nama Nabi dan Rasul berdasarkan dengan kata yang dicari. Dengan penerapan Algoritma Winnowing pada Aplikasi kisah para Nabi dan Rasul dapat mempercepat proses pencarian kisah para Nabi dan Rasul. Aplikasi kisah para Nabi dan Rasul dirancang menggunakan Microsoft Visual Studio 2008 dan MySQL.

REFERENCES

- [1] E. S. A, Wawasan Islam : Pokok-Pokok Pikiran Tentang Pradigma dan Sistem Islam, Jakarta: Gema Insani, 2004.
- [2] D. U. Ade Ambarwati Br Ginting, "PERANCANGAN APLIKASI CATALOG WISATA DI SUMATERA UTARA MENGGUNAKAN ALGORITMA RABIN-KARP," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 57-63, 2019.
- [3] Jarwati, A. C. Prihandoko and W. E. Yulia, "Penerapan Algoritma Winnowing Pada Sistem Rekomendasi Penentuan Dosen Pembimbing Skripsi (Studi Kasus Program Studi Sistem Informasi)," *BERKALA SAINSTEK*, vol. 5, no. 1, pp. 11-20, 2017.
- [4] R. Agus, *Pemodelan Sistem Cetakan Pertama*, Yogyakarta: Graha Ilmu, 2011.
- [5] M. Kusriani, *Tuntunan Praktis Membangun Sistem Informasi Akutansi dengan Visual Basic dan MicrosoftSql Server+Cd*, Yogyakarta: ANDI, 2007.
- [6] D. Wiryantu, *Apl Rekayasa kontruksi Vb6.0+cd*, Jakarta: PT Elex Media komputindo , 2005.
- [7] S. Utama Ema, *Belajar Logika dan Algoritma. Menggunakan bahasa C dan C++ di GNU/Linux*, Yogyakarta: ANDI, 2005.
- [8] Hendrayudi, *vb 2008 Untuk keperluan berbagai programming*, Jakarta: Elex Media Komputindo, 2009.
- [9] H. Jogianto, *Analisa & Desain Sistem Informasi Pendekatan, Terstruktur Teori dan Praktek Aplikasi Bisnis*, Yogyakarta: Andi, 2005.
- [10] Martiani, *101 Info Tentang Kisah Nabi dan Rasul*, Bandung: Dar! Mizan, 2013.
- [11] Khoirotunnisa, *Aplikasi Informasi Kisah 25 Nabi dan Rasul*, Universitas Gunadarma, 2012.
- [12] S. M. S. Rosa Ariani, *Rekayasa Perangkat lunak (Arsitektur dan Berorientasi Objek)*, Bandung: Bandung : Informatika, 2014.
- [13] D. W. & A. A. Schleimes S, "Winnowing Local Algorithms For domkument Fingerprinting," *Journal of SIGMOD*, pp. 1-10, 2003.