

Implementasi Algoritma *Quicksort* dan *Binary Search* pada Fitur Pencarian Media Sosial *Star*

Yoseph RikardusRasi¹, Intan Nur Farida², Ratih Kumalasari Niswatin³

^{1,2,3}Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri

E-mail: ¹richyapc5@gmail.com, ²in.nfarida@gmail.com, ³ratih.workmail@gmail.com

Abstrak –Media sosial dapat digunakan sebagai sarana untuk mencari data, informasi dan pembelajaran secara daring. Fitur Pencarian merupakan salah satu fitur yang lumrah pada sebuah aplikasi media sosial. Dalam kesehariannya selain menggunakan search engine seperti: google dan yahoo, mahasiswa pecinta pemrograman web Universitas Nusantara PGRI Kediri juga menggunakan media sosial dalam mencari informasi. Penelitian ini bertujuan untuk membuat aplikasi media sosial star yang berbasis web sehingga dapat diakses menggunakan media internet. Aplikasi media sosial star ini dibuat untuk membantu mahasiswa pecinta pemrograman web Universitas Nusantara PGRI Kediri. Aplikasi media sosial star ini menyediakan antar muka pengguna yang sederhana sehingga memudahkan user dalam menggunakannya. Dalam pengembangannya aplikasi media sosial star ini dibuat dengan menggunakan metode MVC (Model View Controller). Algoritma pencarian yang digunakan pada pencarian kata adalah menggunakan metode binary search. Pencarian binary search ini dilakukan untuk memperkecil jumlah operasi perbandingan antara data yang akan dicari dengan data yang ada dalam basis data. Dalam prakteknya binary search melakukan pencarian dengan terlebih dahulu membagi menjadi dua data dalam basis data yang sudah terurut terlebih dahulu. Aplikasi ini dapat melakukan pencarian pengguna yang efektif dan informasi yang ditampilkan dalam aplikasi ini sangat akurat.

Kata Kunci— Binary Search, Media Sosial, MVC, Quicksort

1. PENDAHULUAN

Interaksi antar manusia yang melibatkan komunikasi merupakan sesuatu hal yang alamiah yang terjadi di antara manusia. Komunikasi tersebut digunakan untuk saling berhubungan agar penyampaian suatu informasi dapat tersampaikan dengan baik dan jelas. Jejaring social merupakan sebuah struktur sosial yang memberikan kesempatan kepada pengguna untuk memperluas hubungan sosial. Tingginya pengguna jejaring yang bahkan tidak dapat dibatasi cakupannya, dapat dijadikan sebuah kesempatan untuk mempermudah hubungan yang dibutuhkan oleh sebuah organisasi tertentu [1].

Media social telah memungkinkan lahirnya bentuk baru dari organisasi sosial dan interaksi social berbasis jaringan informasi elektronik. Walaupun teknologi informasi tidak secara langsung menyebabkan perubahan sosial, namun teknologi ini merupakan bagian yang tidak terpisahkan dari banyaknya pergerakan dalam perubahan social seperti bentuk baru produksi dan manajemen, adanya media komunikasi baru atau populer disebut sebagai globalisasi ekonomi dan budaya. Media sosial juga telah melahirkan bisnis baru, cara berdagang yang berbeda, bisa menjadi sumber penghasilan bahkan bisa membantu lembaga dalam menjalankan tujuannya [2].

Masalahnya adalah sampai saat ini, belum ada media sosial yang secara spesifik menghubungkan

mahasiswa pecinta pemrograman web Program Studi Teknik Informatika Universitas Nusantara PGRI Kediri (Prodi TI UN PGRI Kediri), untuk berbagi informasi seputar pemrograman web.

Adapun media sosial yang ada seperti : *facebook, instagram, twitter, line, snapchat*, maupun media sosial lokal seperti : *sebangsa.com, oorth, buzz buddies, kwikku* dan *yogrt*, hanya dapat digunakan untuk berhubungan secara daring. Oleh karena itu dibutuhkan sebuah media sosial yang tidak hanya sebatas bersosialisasi atau berhubungan secara daring, tetapi menyediakan fitur pembelajaran khusus untuk mahasiswa pecinta pemrograman web, seperti:

1) Fitur pembelajaran basis data (*Database Learning Feature*).

Fitur pembelajaran basis data menyediakan materi pembelajaran terkait basis data.

2) Fitur pembelajaran desain web (*Web Design Learning Features*).

Fitur pembelajaran desain web ini menyediakan materi pembelajaran terkait Web Design, seperti: *HTML, CSS, JS* dan *framework* untuk desain web seperti: *bootstrap* dan *materialize*.

3) Fitur pembelajaran pemrograman web (*Web Programming Learning Features*).

Pada fitur pembelajaran pemrograman web ini menyediakan materi pembelajaran yang berhubungan dengan *web programming language*, seperti: *JS* dan *PHP*.

Adapun fitur pencarian pengguna adalah fitur yang penting dalam sebuah aplikasi media sosial. Pencarian pengguna lain dalam sebuah aplikasi media sosial adalah hal yang lumrah ketika seorang pengguna ingin menjalin hubungan pertemanan secara daring, mencari pengguna yang ahli dalam bidang tertentu atau hanya sekedar mencari tahu data pengguna lain (*stalking*).

Kecepatan proses pencarian dan keakuratan data adalah masalah berarti dalam menentukan suatu keputusan. Sistem pencarian data menggunakan algoritma yang kurang tepat sering menjadi penyebab utama dari kelambatan proses pencarian maupun ketidakakuratan data yang didapat. Hal ini disebabkan oleh tingkat kemungkinan kelemahan algoritma dalam melakukan pencarian data, terutama ketika data yang tersedia berjumlah sangat besar.

Pada sebuah aplikasi media sosial yang memiliki banyak pengguna, algoritma yang tepat sangat dibutuhkan dalam proses pencarian pengguna hal ini bertujuan agar data yang diperoleh lebih cepat dan akurat.

Dalam penulisan artikel ini penulis menggali informasi dari penelitian-penelitian sebelumnya sebagai bahan perbandingan, baik mengenai kekurangan atau kelebihan yang sudah ada. Adapun penelitian-penelitian yang berkaitan dengan judul yang digunakan untuk memperoleh landasan teori ilmiah, antara lain sebagai berikut:

Penelitian yang berjudul Perancangan Aplikasi Jejaring Sosial Alumni Pecinta Aplikasi Berbasis *ITC* Dengan Algoritma *Graph* Berbasis Web, penelitian ini membahas tentang proses perancangan aplikasi jejaring sosial berbasis web menggunakan algoritma *graph* [1]. Penelitian yang berjudul Penerapan Algoritma Pencarian *Binary Search* dan *QuickSort* pada Aplikasi Kamus Bahasa Palembang Berbasis Web, dalam artikel ini aplikasi kamus yang dibuat merupakan aplikasi yang berbasis web sehingga dapat diakses oleh banyak *user* menggunakan media internet [3]. Penelitian berikutnya yang terkait adalah penelitian yang berjudul Analisa Perbandingan Algoritma *Bubble Sort*, *Merge Sort*, Dan *Quick Sort* Dalam Proses Pengurutan Kombinasi Angka Dan Huruf, penelitian membahas tentang perbandingan algoritma pengurutan *Bubble Sort*, *Merge Sort* dan *Quick Sort* [4].

2. METODE PENELITIAN

Dalam prosedur penelitian ini terdapat beberapa tahapan untuk pengumpulan data, dengan

teknik tertentu. Dalam penelitian ini, metode penelitian yang digunakan adalah sebagai berikut:

2.1 Obyek Penelitian

Obyek penelitian yang digunakan dalam penelitian ini yaitu mahasiswa pecinta pemrograman web Program Studi Teknik Informatika Universitas Nusantara PGRI Kediri tingkat empat (4) sebagai sampel penelitian. Obyek ditentukan berdasarkan hasil wawancara dengan setiap individu (mahasiswa) yang mengaku sebagai pecinta pemrograman web.

2.2 Jenis Data

Jenis data yang digunakan dalam penelitian ini adalah data primer yaitu data yang diperoleh dari mahasiswa pecinta pemrograman web Program Studi Teknik Informatika Universitas Nusantara PGRI Kediri, data tersebut adalah data tentang rancangan atau desain pembuatan aplikasi media sosial *star* dan data pribadi mahasiswa pecinta pemrograman web Program Studi Teknik Informatika Universitas Nusantara PGRI Kediri.

2.3 Metode Pengumpulan Data

Metode yang digunakan dalam proses pengumpulan data sebagai bahan pembuatan sistem adalah wawancara langsung dengan mahasiswa pecinta pemrograman web Program Studi Teknik Informatika Universitas Nusantara PGRI Kediri tingkat empat (4) sebagai sampel penelitian untuk mendapatkan data tentang perancangan aplikasi media sosial *star* yang dibangun dan data pribadi mahasiswa pecinta pemrograman web.

2.4 Metode Pengembangan Sistem

Fase pengembangan sistem ini disebut siklus hidup pengembangan sistem informasi, dengan tahapan-tahapan diantaranya adalah sebagai berikut:

Yang pertama kali dilakukan pada penelitian ini adalah dimulai dengan menentukan kebutuhan kerja yang belum terpenuhi. Menyangkut studi kebutuhan pengguna, studi kelayakan baik secara teknis maupun secara teknologi serta penjadwalan pengembangan proyek sistem informasi.

Tahap kedua (2) adalah menganalisis terkait pertanyaan-pertanyaan penting seperti siapa para pengguna sistem, apa yang akan dicapai oleh sistem dan dimana sistem akan digunakan, serta kapan sistem akan dijalankan.

Selanjutnya proses desain, pada tahap ini ditentukan bagaimana sistem akan bekerja, mempertimbangkan semua perangkat keras, perangkat lunak, infrastruktur jaringan, antar muka pengguna, *form*, *display*, *program*, *database*, dan *file* ditetapkan.

Tahap keempat (4) adalah tahap implementasi sistem, yaitu tahap pemrograman (*coding*). Dalam tahap pemrograman dilakukan implementasi hasil rancangan ke baris-baris kode pemrograman yang dapat dimengerti oleh mesin komputer. Adapun *software* yang digunakan untuk menerjemahkan

kedalam bahasa mesin pada perancangan aplikasi ini adalah *software* : *VS CODE*, *PHP*, *XAMPP* dan *databaseMySQL*.

Tahap pengujian, pada tahap ini berkaitan dengan pemilihan perangkat keras, penyusunan perangkat lunak aplikasi (pengkodean/*coding*), serta pengujian *algoritma* (*quicksort* dan *binary search*) yang diterapkan dalam sistem sudah sesuai dengan kebutuhan atau belum. Adapun yang dilakukan adalah sebagai berikut:

Tahap pertama adalah tahap perencanaan, pada tahap ini dilaksanakan proses identifikasi kebutuhan serta menetapkan metode pembangunan aplikasi media social *star*.

Kedua (2) adalah tahap analisis, pada tahap analisis ini dilaksanakan proses analisis media sosial yang ada dan menetapkan kebutuhan media social *star* yang dibangun.

Berikutnya tahap ketiga (3) adalah tahap desain, pada tahap ini mulai dilakukan pembuatan rancangan aplikasi media social yang dibangun, penentuan proses pada media sosial yang dibangun dan menentukan kebutuhan data dalam pembangunan media social *star*.

Tahap keempat (4) adalah tahap implementasi, pada tahap ini dilakukan implementasi algoritma *quicksort* dan *binary search* pada fitur pencarian media social *star*.

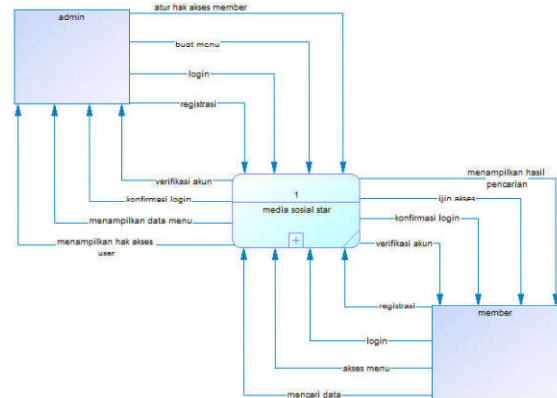
Selanjutnya tahap ke lima (5) adalah tahap pengujian, pada tahap pengujian ini dilakukan pengujian terhadap algoritma *quicksort* dan *binary search* yang diterapkan pada fitur pencarian media social *star*.

3. HASIL DAN PEMBAHASAN

Model view controller (MVC) adalah suatu metode pendekatan yang berorientasi objek yang membagi aplikasi menjadi tiga (3) bagian, yaitu *model* yang mengurus interaksi antara aplikasi dan *database*, *view* yang mengurus urusan logika pemrograman dan *controller* yang mengatur interaksi antara *view* dan *model*.

Untuk memperoleh informasi pengguna secara cepat dan tepat pada aplikasi media sosial yang populer seperti: *facebook* atau *instagram*, biasanya orang mengetikkan nama pengguna pada kotak pencarian yang disediakan. Kemudian sistem akan mencari dan menampilkan hasil pencarian sesuai dengan kata yang ditulis apabila ditemukan, jika tidak maka sistem akan menampilkan pesan bahwa pengguna yang dicari tidak ditemukan. Hal inilah yang akan diterapkan dalam sistem pencarian pengguna pada aplikasi media social *star* yang dibangun.

Untuk desain sistem yang akan dibuat secara umum dapat dilihat pada Gambar 1. *Data flow diagram level 0* berikut ini:



Gambar 1. *Data flow diagram level 0*

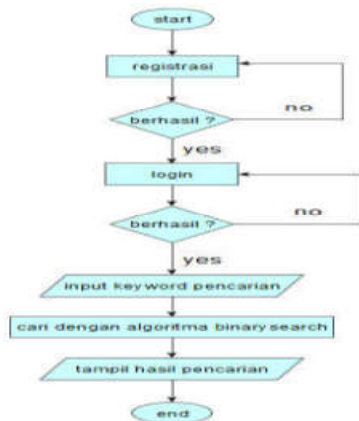
Penjelasan dari Gambar 1. *DFD level 0* adalah sebagai berikut:

- 1) Mula-mula *user* harus melakukan registras iterlebih dahulu.
- 2) Selanjutnya user dapat melakukan login sesuai dengan akun yang sudah terdaftar.
- 3) Sebelumnya sistem membagi *user* ke dalam dua kategori, yakni sebagai *administrator* dan *member*.
- 4) Untuk *user* dengan kategori *administrator* memiliki hak akses penuh terhadap psistem.
- 5) Untuk *user* dengan kategori *member* memiliki hak akses terbatas tergantung pemberian dari *administrator*. Namun ada beberapa fitur bawaan yang diijinkan untuk diakses oleh siapa saja, misalnya: fitur pencarian.

Proses pencarian pengguna yang tersimpan dalam basis data aplikasi media social *star* yang dibangun menerapkan algoritma pencarian *binary search* sehingga menghasilkan waktu yang cepat dan efisien. Aplikasi media social ini dibuat dengan menggunakan bahasa pemrograman web *PHP* dan *database* yang digunakan untuk menyimpan data pengguna adalah *MySQL*.

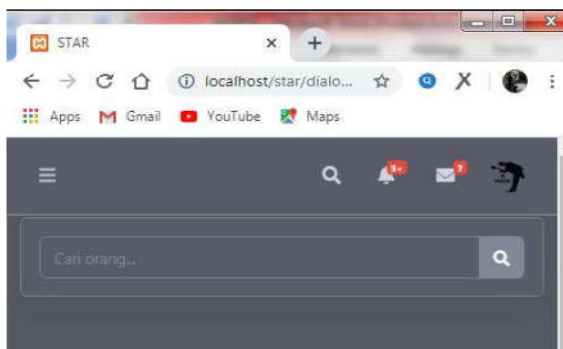
Fitur pencarian sistem ini menyediakan *search box* dalam sistem yang berfungsi bagi *user* dalam menginputkan *keyword* atau kata kunci pencarian, kemudian kata tersebut akan dicari dalam *database* dengan menerapkan algoritma pencarian *binary search*. Algoritma *binary search* diterapkan pada pencarian data dalam tabel ketika pengguna menekan *icon searching*.

Alur proses pencarian pengguna menggunakan algoritma *binary search* dapat dilihat pada Gambar 3 berikut:



Gambar 3. Flowchart sistem pencarian

Tampilan halaman pencarian pengguna pada aplikasi media social berbasis web yang telah dibuat dalam penelitian dapat dilihat pada Gambar 4.



Gambar 4. Halaman pencarian pengguna

Aplikasi ini menyediakan *user interface* yang menyediakan fungsi *input* kata. Kata yang digunakan sebagai *keyword* di *inputkan* user kemudian dicari padanan katanya dalam basis data dengan menggunakan metode *binary search*.

Algoritma pencarian *binary search* akan dijalankan ketika user melakukan pencarian kata. Sebelum proses pencarian kata, terlebih dahulu akan dilakukan pengurutan data yang menerapkan algoritma *quicksort*.

Quicksort merupakan Algoritma Sorting yang dikembangkan oleh C.A.R Hoare pada tahun 1960 yang secara kasus rata-rata, membuat pengurutan $O(n \log n)$ untuk mengurutkan n item. Algoritma ini juga dikenal sebagai *Partition-Exchange Sort* atau disebut sebagai Sorting pergantian pembagi. Pada kasus terburuknya, algoritma ini membuat perbandingan $O(n^2)$, walaupun kejadian seperti ini sangat langka. *Quicksort* sering lebih cepat dalam praktiknya dari pada algoritma $O(n \log n)$ yang lainnya. Dan juga, urutan dan referensi lokalisasi

memori *quicksort* bekerja lebih baik dengan menggunakan cache CPU, jadi keseluruhan sorting dapat dilakukan hanya dengan ruang tambahan $O(\log n)$ [4].

Proses pengurutan data menggunakan algoritma *quicksort* adalah sebagai berikut:

- 1) Penentuan data yang akan diurutkan.
- 2) Memilih elemen pivot, elemen pivot ini sementara yang digunakan sebagai acuan selesainya proses partisi.
- 3) Kemudian memilih elemen pertama dan elemen terakhir.
- 4) Selanjutnya melakukan pertukaran elemen berdasarkan pergerakan dari kiri kekanan dan dari kanan ke kiri sampai ditemukan nilai kiri \geq pivot dan nilai kanan \leq pivot.

Dengan cara ini, seluruh elemen data diurutkan secara rekursif menggunakan algoritma yang sama. Pemilihan pivot atau elemen nilai tengah dalam algoritma *quicksort* sangat menentukan apakah algoritma tersebut menghasilkan performa yang baik atau tidak.

Ada beberapa cara dalam menentukan elemen pivot diantaranya adalah sebagai berikut:

- 1) Menentukan pivot dari elemen yang pertama, terakhir atau elemen tengah dalam tabel. Cara seperti ini hanya baik jika elemen dalam tabel tersusun secara acak.
- 2) Menentukan pivot secara acak dari salah satu elemen tabel.
- 3) Menentukan pivot berdasarkan elemen median dari tabel. Ini adalah cara yang paling tepat karena hasil partisi akan menghasilkan dua bagian tabel yang berukuran seimbang dan kompleksitas waktu yang diberikan minimum.

Pseudocode untuk menentukan pivot *quicksort* adalah sebagai berikut: [3]

```

function partisi(kiri, kanan, pivot)
kiriPointer = kiri
kananPointer = kanan - 1
while Benar do
while X[++kiriPointer] <
pivot do
//do-nothing
end while
while kananPointer > 0 &&
X[--kananPointer] > pivot do
//do-nothing
end while
if kiriPointer >=
kananPointer
break
else
swap kiriPointer,
kananPointer
end if
end while
swap kiriPointer, kanan
return kiriPointer
end function
    
```

Setelah proses penentuan pivot selanjut algoritma *quicksort* dapat dilihat dari *pseudocode* berikut ini: [3]

```

procedure QS(kiri, kanan)
  if kanan-kiri <= 0
    return
  else
    pivot = X[kanan]
  partisi_x = partisi(kiri, kanan, pivot)
  QS(kiri, partisi-1)
  QS(partisi+1, kanan)
end if
end procedure
    
```

Agar lebih memahami *pseudocode* dari algoritma *quicksort* diatas, dapat dilihat contoh berikut ini. Misalnya ada sekumpulan data sebagai berikut:

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
-------	------	------	------	-------	-----	------

a. Langkah pertama adalah tentukan pivotnya., dalam hal ini saya memilih Gina.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			

b. Kemudian buat partisi buat masing-masing angka sebelah kanan dan kiri.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
Partisi 1 [i]			pivot	Partisi 2 [j]		

c. Kemudian gunakan algoritma *quicksort* yang ada diatas, jika kata lebih kecil dari pivot maka akan diletakan sebelah kiri dan jika lebih besar maka letakan disebelah kanan. Langkah pertama adalah bandingkan Ipang dengan pivot apakah lebih kecil atau lebih besar.

↓			↓			
Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
				Gina		
				Pivot		

d. Karena Ipang lebih besar maka letakan Ipang setelah pivot.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
				Gina	Ipang	
				Pivot		

e. Lanjut ke angka Dedy, bandingkan angka Dedy dengan pivot.

	↓		↓			
Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
				Gina	Ipang	
				Pivot		

f. Karena angka Dedy lebih kecil dari Gina maka posisi tetap.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy				Gina	Ipang	
				Pivot		

g. Lanjut ke Budi, cek apakah Budi lebih kecil atau lebih besar dari pivot.

		↓	↓			
Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy				Gina	Ipang	
				Pivot		

h. Karena Budi lebih kecil dari pivot maka letaknya tetap.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy	Budi			Gina	Ipang	
				Pivot		

i. Bandingkan pivot dengan Julio, cek Julio lebih besar atau lebih kecil dari pivot.

			↓	↓		
Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy	Budi			Gina	Ipang	
				Pivot		

j. Karena angka 10 lebih besar maka posisi tetap sebelah kanan.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy	Budi			Gina	Ipang	Julio
				Pivot		

k. Lanjut ke Ayu, cek Ayu lebih kecil atau lebih besar dari pivot.

			↓		↓	
Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy	Budi			Gina	Ipang	Julio
				Pivot		

l. Karena lebih kecil maka pindah kesebelah kiri pivot.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy	Budi	Ayu		Gina	Ipang	Julio
				Pivot		

m. Lanjut ke Elma, cek apakah Elma lebih kecil atau lebih besar dari pivot.

			↓			↓
Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy	Budi	Ayu		Gina	Ipang	Julio
				Pivot		

n. Karena Elma lebih kecil maka pindah ke sebelah kiri pivot.

Ipang	Dedy	Budi	Gina	Julio	Ayu	Elma
			Pivot			
Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
				Pivot		

o. Setelah itu masuk ke dalam partisi baru, sampai disini proses belum selesai.

Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
Partisi 1 [i]			Partisi 2 [j]			

p. Tentukan pivot untuk masing-masing partisi.

Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
pivot						pivot

q. Perbandingan untuk pivot pertama, Budi, cek apakah Budi lebih kecil atau lebih besar dari pivot.

	↓					
Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
Pivot						Pivot
		Dedy				Julio
		Pivot				Pivot

r. Karena Budi lebih kecil dari pivot maka pindahkan ke kiri pivot.

Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
Pivot						Pivot
	Budi	Dedy				Julio
		Pivot				Pivot

s. Lanjut ke Ayu, cek apakah Ayu lebih besar atau lebih kecil dari pivot.

↓		↓				
Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
Pivot						Pivot
	Budi	dedy				Julio
		Pivot				Pivot

t. Karena Ayu lebih kecil maka pindahkan di sebelah kiri angka pivot.

Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
Pivot						Pivot
Ayu	Budi	Dedy				Julio
		Pivot				Pivot

u. Lanjut ke Elma, cek apakah Elma lebih besar atau lebih kecil dari pivot.

			↓			
Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
Pivot						Pivot
Ayu	Budi	Dedy				Julio
		Pivot				Pivot

v. Karena lebih besar maka posisinya tetap, untuk partisi pertama selesai.

Dedy	Budi	Ayu	Elma	Gina	Ipang	Julio
Pivot						Pivot
Ayu	Budi	Dedy	Elma			Julio
		Pivot				Pivot

w. Ulangi langkah-langkah seperti sebelumnya untuk pivot partisi ke 2. dan hasil akhir dari *quicksort* ini adalah seperti berikut ini:

Ayu	Budi	Dedy	Elma	Gina	Ipang	Julio
-----	------	------	------	------	-------	-------

Adapun proses pencarian menggunakan algoritma *binary search* dapat dilihat pada *pseudocode* berikut: [3]

```

Procedure pencarian_biner
  X ← sorted array
  Y ← size of array
  Z ← value to be searched
  Set data_low = 1
  Set data_up = Y
  while Z not found
    if data_up < low
      EXIT: Z does not exists.
    Set mid_point = low + (data_up - data_low) / 2
    if X[mid_point] < Z
      set data_low = mid_point + 1
    if X[mid_point] > Z
      set data_up = mid_point - 1
    if X[mid_point] = Z
      EXIT: Z found at location mid_point
  end while
end procedure

```

Penjelasan dari *pseudocode binary search* dapat dijelaskan dalam langkah-langkah berikut:

a. Mula-mula ditentukan batas bawah ($data_low$) = 1 dan batas atas ($data_up$) = N dari *array* yang sudah terurut.

- b. Cari posisi data tengah (*mid_Point*) dengan rumus: $\text{data_low} + (\text{data_up} - \text{data_low})/2$.
- c. Data (*Z*) yang dicari dibandingkan dengan data tengah (*mid_Point*).
- d. Jika lebih kecil, proses dilakukan kembali tetapi *data_up* dianggap sama dengan posisi tengah - 1 { $\text{data_up} = \text{mid_Point} - 1$ }.
- e. Jika lebih besar, proses dilakukan kembali tetapi *data_low* dianggap sama dengan posisi tengah + 1 { $\text{data_low} = \text{mid_Point} + 1$ }.
- f. Demikian seterusnya sampai data tengah sama dengan data yang dicari ($\text{mid_Point} = Z$).

Agar lebih memahami penjelasan *pseudocode binary search* diatas, dapat dilihat pada contoh berikut. Misalkan akan dicari pengguna dengan nama Ipang pada sekumpulan data berikut:

{Ayu, Budy, Dedy, Elma, Gina, Ipang, Julio, Kerin, Leon, Richi}

- a. Mencari data tengah dengan rumus $\text{data_mid} = \text{data_low} + (\text{data_high} - \text{data_low})/2$ $\text{data_mid} = 0 + (9 - 0)/2 = 4$ maka akan didapat data tengah adalah Gina.
- b. Kemudian data yang dicari Ipang dibandingkan dengan data tengah (*data_mid*) yang diperoleh yaitu Gina.
- c. Hasil perbandingan $\text{Ipang} > \text{Gina}$ maka set $\text{data_low} = \text{data_mid} + 1$, $\text{data_mid} = \text{data_low} + (\text{data_high} - \text{data_low})/2 = 7$, Nilai *data_mid* sekarang adalah Kerin.
- d. Kemudian data yang dicari Ipang dibandingkan dengan data tengah (*data_mid*) yang peroleh yaitu Kerin.
- e. Hasil perbandingan $\text{Ipang} < \text{Kerin}$ maka set $\text{data_low} = \text{data_mid} - 1$, set $\text{data_up} = \text{data_mid} - 1$, $\text{data_mid} = \text{data_low} + (\text{data_up} - \text{data_low})/2 = 6$, Nilai *data_mid* sekarang adalah Julio.
- f. Kemudian data yang dicari Ipang dibandingkan dengan data tengah (*data_mid*) yang diperoleh yaitu Julio.
- g. Hasil perbandingan $\text{Ipang} < \text{Julio}$ maka set $\text{data_low} = \text{data_mid} - 1$, set $\text{upperBoud} = \text{data_mid} - 1$, $\text{data_mid} = \text{data_low} + (\text{data_up} - \text{data_low}) = 5$, nilai *mid* sekarang adalah Ipang.
- h. Kemudian data yang dicari Ipang dibandingkan dengan data tengah (*data_mid*) yang diperoleh yaitu Ipang.
- i. Hasil perbandingan $\text{Ipang} = \text{Ipang}$, data ditemukan di posisi 5, maka perulangan berhenti.

Agar sistem dapat melakukan pencarian dan menampilkan hasil pencarian pengguna dengan cepat dan tepat dibutuhkan sebuah proses yang benar. Untuk itu pada fitur pencarian pengguna pada aplikasi media sosial yang dibangun diterapkan

algoritma *Quicksort* dan *Binary Search* untuk melakukan proses pencarian.

Pencarian *binary search* ini dilakukan untuk memperkecil jumlah operasi perbandingan antara data yang akan dicari dengan data yang ada dalam tabel. Prinsipnya melakukan pembagian ruang pencarian secara berulang-ulang sampai data yang cari ditemukan atau sampai tidak dapat dibagi lagi dengan kata lain data yang dicari tidak ditemukan [4].

Pengujian Sistem

Tujuan dilakukannya pengujian terhadap sistem adalah untuk memastikan apakah sistem telah memiliki kemampuan yang baik sesuai dengan tujuan perancangan atau belum, yaitu menampilkan pengguna sebagai hasil pencarian berdasarkan *keyword* yang diketikan pada *search box*.

a. Rencana Pengujian Sistem

Adapun rencana pengujian sistem yang akan diuji dapat dilihat pada Tabel 2 berikut.

Tabel 2. Rencana pengujian sistem

No	Komponen sistem yang diuji	Butir uji
1	Fitur pencarian	Sistem pencarian harus mampu menampilkan pengguna sesuai dengan <i>keyword</i> yang diinputkan pada <i>search box</i> dan kemampuan sistem pencarian dalam menampilkan pengguna harus berdasarkan perhitungan dari algoritma pencarian <i>binary search</i>

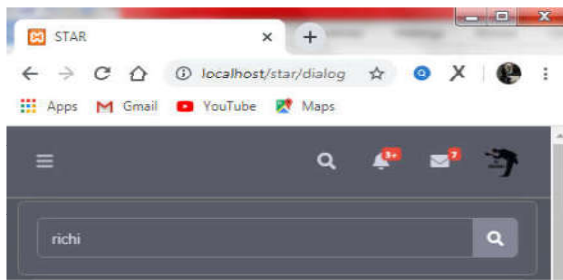
b. Hasil Pengujian Sistem

Pada Tabel 3 akan dijabarkan hasil pengujian sistem yang telah dilakukan pencarian.

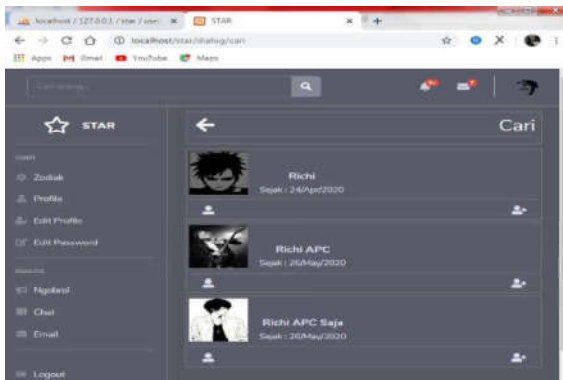
Tabel 3. Hasil pengujian sistem

No	Skenario uji	Hasil yang diharapkan	Hasil pengujian
1	Menampilkan pengguna pada pengetikan nama pengguna, yaitu nama yang diinputkan pada <i>search box</i> sama dengan nama yang tersimpan pada <i>database</i> baik itu nama panggilan maupun nama lengkap	Sistem menampilkan pengguna dengan nama yang sesuai dengan kata yang diketikan	Berhasil

Untuk tampilan *input* dan *output* data pada fitur pencarian sistem dapat dilihat pada Gambar 5. *Input* kata pencarian, dan Gambar 6. *Output* pencarian berikut ini:



Gambar 5. Input kata pencarian



Gambar 6. Output pencarian

Dari Gambar 6. Output pencarian terlihat bahwa sistem menampilkan semua akun untuk pengguna yang nama akunnya terdapat kata *richi* dan harus ada akun yang nama penggunanya hanya terdiri dari satu kata yaitu kata *richi* saja sesuai dengan *keyword* pencarian.

4. SIMPULAN

Dari pembahasan yang telah diuraikan maka penulis mencoba membuat kesimpulan sebagai berikut:

- 1) Penerapan algoritma pencarian kata dalam basis data dapat menggunakan teknik *binary search*.
- 2) Untuk mempermudah menerapkan algoritma pencarian *binary search* yang mengharuskan data dalam keadaan terurut terlebih dahulu, maka digunakan teknik pengurutan *quick sort*.
- 3) Metode *quick sort* sangat tepat digunakan untuk mengurutkan data pada *database*.

5. SARAN

Sistem ini masih belum sempurna, sehingga perlu dikembangkan untuk memperbaiki kinerja sistem. Adapun saran-saran dari penulis adalah sebagai berikut:

- 1) Sistem yang dibuat adalah sistem yang berbasis web dan dapat dikembangkan untuk sistem android.
- 2) Sistem ini dapat dikembangkan untuk seluruh mahasiswa Fakultas Teknik Universitas

Nusantara PGRI Kediri. Misalnya untuk mahasiswa pecinta robot dibuatkan fitur pembelajaran untuk merakit robot dan untuk mahasiswa pecinta mesin dibuatkan fitur pembelajaran untuk merakit mesin kendaraan.

- 3) Sistem ini dapat dikembangkan untuk mencari pengguna berdasarkan nama depan, nama tengah maupun nama belakang pengguna agar proses pencarian lebih fleksibel.

DAFTAR PUSTAKA

- [1] Latifah, F., dan Saefudin, I. 2018. Perancangan Aplikasi Jejaring Sosial Alumni Pecinta Aplikasi Berbasis ITC Dengan Algoritma Graph Berbasis WEB. *Jurnal Petir*. No. 2. Vol. 11. 1978-9262.
- [2] Anwar, K.R., dan Rusmana, A. 2017. Komunikasi Digital Berbentuk Media Sosial Dalam Meningkatkan Kompetensi Manaja Sosial Kepala Pustakawan, Dan Tenaga Pengelola Perpustakaan. *Jurnal Aplikasi Ipteks untuk Masyarakat*. No. 3. Vol. 6. 204 – 208.
- [3] Andri. 2019. Penerapan Algoritma Pencarian Binary Search dan Quick Sort pada Aplikasi Kamus Bahasa Palembang Berbasis Web. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, No.01. Vol.04. 2477-5126, https://ejournal.pontitek.ac.id/index.php/informatika/article/view/1104/pdf_31 diakses pada tanggal 15 Januari 2020.
- [4] Sonita, A., dan Nurtafika, F., 2014. Analisis Perbandingan Algoritma Bubble Sort, Merge Sort, Dan Quick Sort Dalam Proses Pengurutan Kombinasi Angka Dan Huruf. *Jurnal Pseudocode*. Nomor 2. Volume II. 2355 – 5920.