uSocial *REALTIME* BERBASIS ANDROID MENGGUNAKAN VOLLEY DAN ALGORITMA *BRUTEFORCE*

Azizah[1], Fauziah[2], Nur Hayati[3]
Informatika, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional[1,2,3]
azizahif99@gmail.com[1]*, fauziah@civitas.unas.ac.id[2], nurhayati@civitas.unas.a.c.id[3]

Submitted October 28, 2020; Revised November 25, 2020; Accepted November 27, 2020

Abstrak

Komunikasi dapat dilakukan dimana saja dan kapanpun sehingga mempermudah masyarakat untuk bertukar informasi. Berkomunikasi saat ini sangat mudah dengan memanfaatkan beragam aplikasi atau media dengan menggunakan fasilitas internet yaitu aplikasi seperti *Instagram, Facebook, WhatsApp Messenger* dan lainnya. Aplikasi uSocial dapat menghubungkan pengguna dalam berkomunikasi, bertukar informasi dan berbagi ilmu yaitu aplikasi uSocial. Aplikasi ini dibuat dengan menerapkan metode *Firebase*, *Volley* dan *RecyclerView*. Ketiga metode tersebut berguna untuk menjalankan aplikasi secara langsung atau *Real-time*. Pengujian algoritma dilakukan dengan membandingkan algoritma *Brute Force* dengan algoritma *Horspool* dan *Knuth Morris Pratt*. Hasil dari pengujian algoritma *Brute Force* pada pencarian kata dilakukan berdasarkan panjangnya kalimat pencarian menghasilkan akurasi sebesar 100% untuk uji coba 500 data serta mempersingkat waktu untuk pencarian *postingan* maka lebih baik menggunakan algoritma *Brute Force*. Pada hasil pengujian menggunakan metode *whitebox* yang diuji menggunakan 3 parameter yaitu *cyclomatic complex city, region , dan independent path* menghasilkan nilai yang sama sebesar 17. Maka alur dan logika program sudah sesuai dengan standar ANSI dan tidak perlu mengubah alur atau program kembali. Sedangkan hasil pada pengujian *blackbox*, fungsi-fungsi yang ada pada aplikasi menghasilkan *output* yang diinginkan.
Kata kunci: Android, *Firebase Real-time*, Media sosial, Aplikasi, *Bruteforce.*

Abstract

Communication can be done anywhere and anytime, making it easier for people to exchange information. Communicating today is very easy by utilizing various applications or media by using internet facilities such as Instagram, Facebook, WhatsApp Messenger, and others. The author builds an application that can connect users in communicating, exchanging information, and sharing the science that is the uSocial application. The app is created by applying the Firebase, Volley, and RecyclerView methods. These three methods are useful for running applications directly or In Real-time. Algorithm testing is done by comparing the Brute Force algorithm with the Horspool, and Knuth Morris Pratt algorithms. The results of Brute Force algorithm testing on word searches performed based on the length of search sentences resulting in 100% accuracy for 500 data trials as well as shortening the time for posts searches then better using Brute Force algorithms. The test results using the Whitebox method tested using three parameters, namely cyclomatic complex city, region, and independent path, produce 17. The program's flow and logic follow ANSI standards and there is no need to change the flow or plan back. At the same time, the results on the Blackbox testing of the functions in the application managed to produce the desired output.
Keywords: *Android, Firebaes Real-time, Social media, Apps, BeruteForce*

1. PENDAHULUAN

Komunikasi memainkan peran yang sangat penting untuk berinteraksi dengan yang lainnya. Dewasa ini berkomunikasi dapat memanfaatkan beragam aplikasi atau media, kalangan masyarakat tidak asing dengan *smartphone* yang dapat menghubungkan satu sama lain dengan menggunakan fasilitas internet dan aplikasi seperti *WhatsApp Messenger, Instagram, Facebook* dan lainnya. Android merupakan sistem operasi untuk ponsel yang dikem-

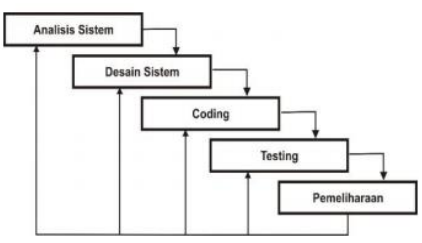
bangkan oleh Google. *Firebase* merupakan *database* yang menggunakan soket yang memungkinkan pengguna untuk menyimpan dan mengambil data dari *database* [1]. *Push notification* merupakan layanan yang digunakan untuk pemberitahuan melalui pesan yang ada di *smartphone* [2]. Dalam layanan ini pemberitahuan atau notifikasi dilakukan secara langsung (*Real-time*) yang mempermudah pengguna dalam menerima informasi atau data yang diperlukan. *Volley* merupakan *library* yang digunakan untuk pertukaran data dari server dengan client dengan lebih mudah dan cepat [3].

Media daring yang digunakan pengguna agar berpartisipasi dengan mudah biasa disebut dengan media sosial [4]. Fitur di berbagai media sosial sangat bervariasi diantaranya berbagi pengalaman, berbagi *postingan*, pesan dan masih banyak lagi. Penelitian yang dilakukan oleh [5] adalah membuat aplikasi media sosial dengan membuat fitur pesan. Pada hasil penelitian yang dilakukan oleh [6] pada tahun 2018 yaitu mengimplementasikan menu pesan dan pengingat tugas. Pada penelitian yang dilakukan oleh [7] mengimplementasikan Algoritma AES pada aplikasi media sosial yang menerapkan fitur beranda dan menu pesan. Pada penelitian yang dilakukan oleh [8] pada tahun 2017 membuat aplikasi media tanya jawab dengan menerapkan membagikannya ke pengguna melalui fitur beranda secara langsung. Kemudian pada penelitian yang dilakukan oleh [9] pada tahun 2018 yang mengimplementasikan fitur notifikasi. Penelitian yang dilakukan oleh [10] pada tahun 2019 membuat aplikasi media sosial yang dapat membuat *postingan* dan *chat messenger* [10].

Pada penelitian ini, penulis menerapkan *volley* dan algoritma *bruteforce* pada aplikasi uSocial. Pada penelitian sebelumnya aplikasi media sosial belum menerapkan keseluruhan menu di dalam aplikasi yang telah dibuat, penulis bermaksud untuk melakukan perbaikan dengan membuat menu pesan, *postingan*, *push notification*s, notifikasi komentar, suka pada *postingan*, dan fitur mencari *postingan* secara langsung. Aplikasi uSocial merupakan aplikasi media sosial berbasis Android dengan menu yang mudah digunakan dan dapat digunakan pada *smartphone* android tanpa menggunakan media komputer lain ataupun *website* untuk saling terhubung satu sama lain baik komunitas, individu ataupun alumni untuk berbagi informasi dan komunikasi.

2. METODE PENELITIAN

Metode penelitian dilakukan dalam beberapa tahap yaitu studi literatur, perumusan masalah, pengembangan sistem dan penarikan simpulan. Kumpulan kegiatan mengenai metodologi pengumpulan data acuan, membaca, mencatat dan mengolah bahan yang akan dilakukan penelitian disebut dengan studi literatur [11]. Dapat mengungkapan teori yang tepat dengan permasalahan yang sedang diteliti sebagai bahan acuan pembahasan penelitian. Perumusan masalah memperjelas masalah sehingga mempermudah dalam menyelesaikannya. Dari hasil studi literatur, penulis menemukan permasalahan yang dapat dirumuskan yaitu, bagaimana membangun aplikasi uSocial dengan menerapkan algoritma *bruteforce* dan beroperasi secara langsung. Tahap pengembangan sistem memakai metode air terjun atau *waterfall* dengan tahapan yang dapat dilihat pada gambar 1 [12].



Gambar 1. Metode *Waterfall*

Untuk mencapai target suatu aplikasi dengan cara mendesain aplikasi untuk memeriksa apakah mencapai kebutuhan atau tidak [13]. Pengembangan sistem dengan melakukan Analisa kebutuhan sistem, desain, pembuatan program dan pengujian aplikasi.

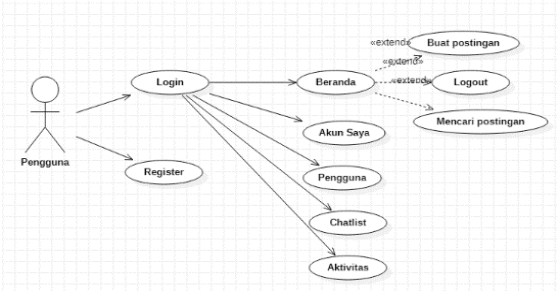
Analisa Kebutuhan Sistem

Analisis sistem didefiniskan sebagai proses dalam mengumpulkan, menginterpretasi-kan kenyataan, mendiagnosa permasalahan dan menyatukannya untuk memperbaiki system [14]. Kebutuhan sistem yang diperlukan dalam membuat aplikasi ini yaitu:

1. Perangkat Keras (*Hardware*)
   Perangkat keras pada penelitian kali ini terdiri dari :
   - Laptop (ASUS VivoBook A442U) dengan spesikasi sebagai berikut
   a. CPU Intel Core i5 8th Gen
   b. RAM 8GB DDR4
   c. NVDIA 930MX
   d. HDD 1TB
2. Perangkat Lunak (*Software*)
   Perangkat lunak yang digunakan pada penelitian terdiri dari :
   - Android Studio 3.4
   -Windows 10 64 bit (Sistem Operasi)

Desain sistem

Pada aplikasi uSocial memiliki fitur yang dapat dilihat pada pada *use case* diagram sebagai berikut:



Gambar 2. *Use case* Diagram Aplikasi uSocial

Pada gambar 2 menjelaskan pengguna dapat mengakses aplikasi uSocial dengan melakukan *register* dan *login*. Setelah masuk kedalam aplikasi pengguna dapat membuat *postingan*, berkomunikasi melalui *chat messenger*, mendapatkan notifikasi jika pengguna lain menyukai dan berkomentar pada *postingan* yang dibuat.

Pengkodean (*coding*)

Aplikasi uSocial ini menerapkan Bahasa pemrograman Java Android dengan mengimplementasikan POJO mengguna-kan *Volley* dalam mengambil dan mengirimkan data dan *RecyclerView* dalam menampilkan *postingan*, isi pesan dan *chatlist*. Selanjutnya algoritma *Brute Force* yang diimplementasikan dalam memban-dingkan *string* dengan tiap karakter. Proses yang dilakukan yaitu membandingkan karakter dimulai dari kiri ke kanan secara berulang sampai menemukan kecocokan antara *string* dan teks [15].

Pengujian (*Testing*)

Blackbox dan *whitebox* merupakan metode yang digunakan untuk pengujian aplikasi uSocial.

Pemeliharaan

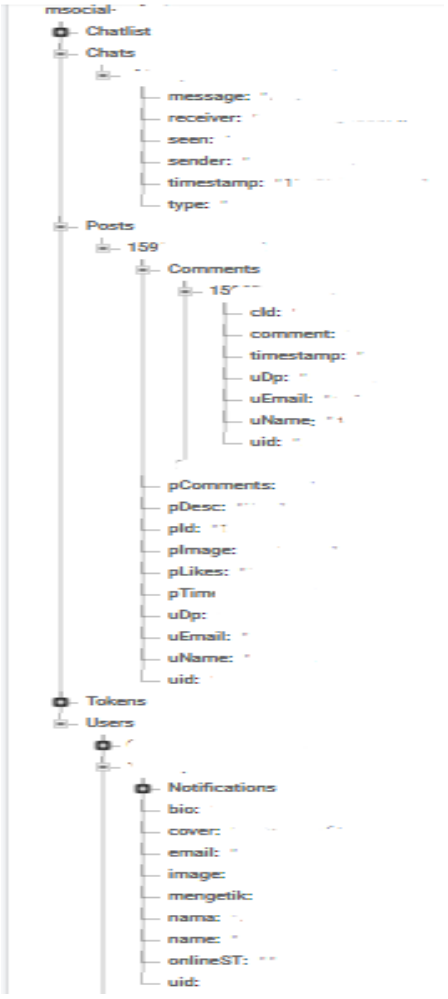
Aplikasi ini akan dilakukan perbaikan setelah adanya evaluasi dari pengguna dengan tujuan agar aplikasi lebih menarik lagi.

3. HASIL DAN PEMBAHASAN

Aplikasi uSocial ini mengimplementasikan *Firebase Authentication* pada saat pengguna mendaftarkan akun dan masuk kedalam aplikasi uSocial. Kemudian mengimplementasikan *Firebase Database Realtime* yang digunakan untuk menyimpan dan mengirimkan data yang diperlukan pengguna. Menerapkan notifikasi dan pesan bisa menggunakan *Cloud Messaging*. Untuk menyimpan data foto yang diunggah oleh pengguna menggunakan *Firebase Storage*.

Struktur *database*

Aplikasi uSocial memiliki struktur *database* yaitu *token*s, *users*, *chatlist*, *post*ss, dan *chat*s. Pada *child users* berisi data akun pengguna yang terdaftar, *child token*s berisi *token* untuk masuk kedalam aplikasi yang dibaca oleh sistem agar selalu terkoneksi dengan *database*. *Token*s ini dibuat pada saat pengguna memasang aplikasi dan mendaftarkan akun pertama kali pada aplikasi uSocial. Data pesan yang dikirimkan oleh pengguna disimpan dalam *child chat*s, *child chatlist* diinisialisasikan untuk menyimpan pesan terakhir dan menampilkannya di menu *chatlist*, dan *child post*ss berisi data yang tersimpan oleh pengguna ketika membuat tulisan atau mengunggah gambar di *home*. Gambar 3 merupakan struktur *database* aplikasi uSocial.



Gambar 3. Struktur *Database*

Struktur Aplikasi

Aplikasi uSocial menggunakan struktur aplikasi yaitu *client-server*, dimana pengguna dapat mengakses data bersamaan dengan *server*.

1. *Client*
   Database Firebase diintegrasikan pada aplikasi uSocial untuk menyimpan serta menampilkan data pengguna secara langsung pada aplikasi.
2. *Server*
   Server bersifat langsung (*Realtime*) yang artinya jika ada perubahan data pada aplikasi maupun *server* otomatis akan ter-*update* pada perangkat pengguna.

Algoritma yang digunakan

Firebase Authentication diterapkan pada aplikasi uSocial agar pada saat membuat akun, data yang disimpan terenkripsi dan aman sehingga tidak ada yang mengetahui kata sandi pengguna dengan menggunakan algoritma seperti ini.

```
Program Firebase Authentication
Private FirebaseAuth penghubung;
Penghubung = FirebaseAuth.getInstance();
penghubung.membuatuser(email, password)
      .addOnCompleteListener(this,      new
      OnCompleteListener<AuthResult>(){
      @Override
      Public    void    onComplete(@NonNull
      Task<AuthResult> task){
        if(task.isSuccessful())
           FirebaseUser      pengguna      =
           penghubung.getCurrentUser();
        end if
        else
        Toast "Authentication failed".
   end for
```

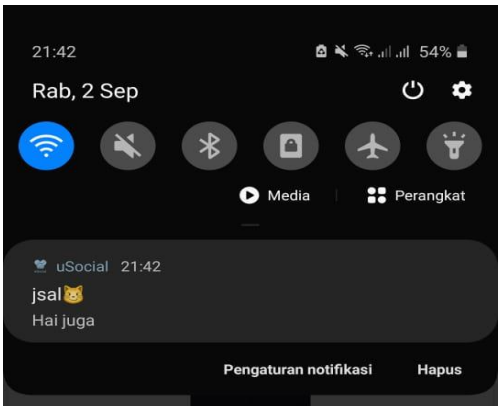
Pada algoritma program yang digunakan untuk membuat akun pengguna menggunakan *Method* penghubung yang memprogram jika berhasil mendaftarkan akun maka pengguna langsung masuk kedalam aplikasi. Namun jika gagal maka akan tampil pesan *error*.

Volley digunakan untuk mengambil data dari *client-server* kemudian menampilkan notifikasi pada perangkat pengguna pada saat mengirimkan atau menerima pesan dengan struktur sebagai berikut.

```
Program push notification
to addbhdbdhbdhd;
notification
    body Hai,
    title Jsal,
    content_available true
    sound default
    priority high

data
    body Hai,
    title Jsal,
    content_available true
    priority high
```

Program *push notification* mengeluarkan *output* berupa notifikasi yang diterima pengguna.

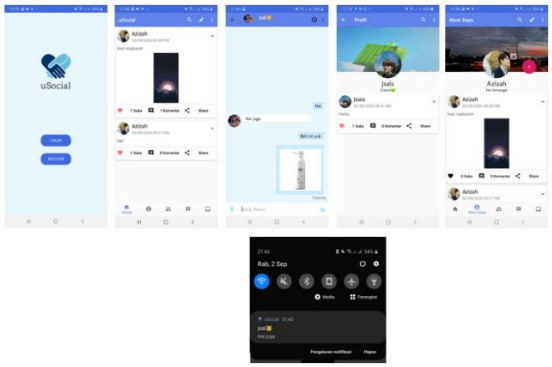


Gambar 4. Tampilan Notifikasi Pesan

Hasil *output* dari program *push notification* dapat dilihat pada gambar 4.

Tampilan Aplikasi

Tampilan aplikasi ini dibuat menggunakan pemrograman XML serta mengimplementasikan struktur *Model-View* dan ditampilkan menggunakan fitur *RecyclerView*. Tampilan keseluruhan aplikasi uSocial dapat dilihat pada gambar 5.



Gambar 5. Tampilan Aplikasi

Aplikasi uSocial memiliki menu *login* dan *register* yang betujuan untuk mendapatkan akun. Selain menu *login* aplikasi ini memiliki fitur menu akun saya, *home*, pesan, aktivitas dan pengguna. Aplikasi ini mengimplementasikan fitur *push notification* pada saat pengguna mengirimkan pesan sehingga pengguna lain dapat menerima notifikasi diperangkatnya.

Penerapan Algoritma *bruteforce*

Penerepan algoritma *bruteforce* yaitu pada saat mencari isi atau konten *postingan*, fitur pencarian ini dibuat untuk mempermudah pengguna dalam mencari konten pengguna lain dengan mudah.

```
Program Search posts
  DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReference("Postss");
        Query query = databaseReference.orderByChild("uid").equalTo(uid);
        query.addValueEventListener(new ValueEventListener() {
            @Override
            Public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

                datapostsList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()){
                    Dataposts dataposts = ds.getValue(Dataposts.class);

                    if (dataposts.getpDesc().toLowerCase().contains(searchqueku.toLowerCase())){

datapostsList.add(dataposts);

                    }

                    adapterposts = new Adapterposts(getActivity(), datapostsList);

postsrv.setAdapter(adapterposts);

                }
            }
```

Program *search posts* ini diterapkan pada menu *home*, akun saya dan pengguna. Tujuan membuat program ini memudahkan pengguna yang ingin mencari pengguna lain atau *postingan* yang menarik dengan mudah.

Gambar 6. Fitur Mencari *Postsing*

Pengguna mencari *postingan* yang berdeskripsi kata "Hai malam" dengan *string* "malam" terlihat pada gambar 6. Langkah pertama dalam implementasi algoritma *Bruteforce* pada pencarian *string* "malam" dapat dilihat pada tabel 1.

Tabel 1. Langkah 1

Deskripsi	H	a	i		m	a	l	a	m
String	m	a	l	a	m				
Indeks	0	1	2	3	4	5	6	7	8

Pada tabel 1 melakukan pencocokan karakter pertama antara *string* dengan deskripsi. Bila belum menemukan kecocokan maka tetap melakukan proses pencocokan.

Tabel 2. Langkah 2

Deskripsi	H	a	i		m	a	l	a	m
String		m	a	l	a	m			
Indeks	0	1	2	3	4	5	6	7	8

Tabel 2 terlihat bahwa karakter a tidak cocok dengan karakter m maka dilanjutkan ke proses pencocokan berikutnya.

Tabel 3. Langkah 3

Deskripsi	H	a	i		m	a	l	a	m
String			m	a	l	a	m		
Indeks	0	1	2	3	4	5	6	7	8

Tabel 3 terlihat bahwa karakter i tidak cocok dengan karakter m maka dilanjutkan ke proses pencocokan berikutnya.

Tabel 4. Langkah 4

Deskripsi	H	a	i		m	a	l	a	m
String				m	a	l	a	m	
Indeks	0	1	2	3	4	5	6	7	8

Pada tabel 4 terlihat bahwa karakter " " atau spasi tidak cocok dengan karakter m maka dilanjutkan ke proses pencocokan berikutnya.

Tabel 5. Langkah 5

Deskripsi	H	a	i		m	a	l	a	m
String					m	a	l	a	m
Indeks	0	1	2	3	4	5	6	7	8

Pada tabel 5 antara deskripsi dengan *string* mengalami kecocokan di indeks ke-4 maka tidak melakukan pergeseran kembali.

Tabel 6. Hasil Pengujian Akurasi Algoritma *Bruteforce* pada Deskripsi dengan S*tring*

No	*Input*	*Output*	Hasil
1	orang-orang	orang-orang	100% Akurat
2	Sejarah	sejarah	100% Akurat
3	Cara	cara	100% Akurat
4	Seni	seni	100% Akurat
5	Dunia	dunia	100% Akurat
6	Informasi	informasi	100% Akurat
7	Peta	peta	100% Akurat
8	Dua	dua	100% Akurat
9	Keluarga	keluarga	100% Akurat
10	Pemerintah	pemerintah	100% Akurat
11	kesehatan	kesehatan	100% Akurat
12	sistem	sistem	100% Akurat
13	komputer	komputer	100% Akurat
14	daging	daging	100% Akurat
15	tahun	tahun	100% Akurat
16	Terima kasih	Terima kasih	100% Akurat
17	musik	musik	100% Akurat
18	orang	orang	100% Akurat
19	bacaan	bacaan	100% Akurat
20	metode	metode	100% Akurat

21	data	data	100% Akurat
22	makanan	makanan	100% Akurat
23	pengertian	pengertian	100% Akurat
24	teori	teori	100% Akurat
25	hukum	hukum	100% Akurat
26	burung	burung	100% Akurat
27	literatur	literatur	100% Akurat
28	masalah	masalah	100% Akurat
29	perangkat lunak	perangkat lunak	100% Akurat
30	kontrol	kontrol	100% Akurat

Tabel 6 merupakan hasil pengujian dengan data deskripsi sebanyak 30 dari 500 data yang memiliki tingkat akurasi kecocokan sebesar 100% pada pencarian *postingan* menggunakan algoritma bruteforce.

Membandingkan dengan Algoritma *Knuth Morris Pratt* dan Algoritma *Horspool*

Pada algoritma *knuth morris pratt* dalam mencari kata dilakukan dengan pencocokan karakter jika belum cocok maka akan melakukan pergeseran sampai kata deskripsi dengan *string* cocok [16]. Gambar 7 merupakan langkah-langkah dalam penerapan algoritma *knuth morris pratt* pada pencarian *string* "malam".

Algoritma *knuth morris pratt*

Tabel 7. Langkah 1

Deskripsi	H	a	i		m	a	l	a	m
String	m	a	l	a	m				
Indeks	0	1	2	3	4	5	6	7	8

Tabel 8. Langkah 2.

Deskripsi	H	a	i		m	a	l	a	m
String		m	a	l	a	m			
Indeks	0	1	2	3	4	5	6	7	8

Tabel 9. Langkah 3.

Deskripsi	H	a	i		m	a	l	a	m
String			m	a	l	a	m		
Indeks	0	1	2	3	4	5	6	7	8

Tabel 10. Langkah 4.

Deskripsi	H	a	i		m	a	l	a	m
String				m	a	l	a	m	
Indeks	0	1	2	3	4	5	6	7	8

Tabel 11. Langkah 5.

Deskripsi	H	a	i		m	a	l	a	m
String					m	a	l	a	m
Indeks	0	1	2	3	4	5	6	7	8

Tabel 12. Langkah 6.

Deskripsi	H	a	i		m	a	l	a	m
String					m	a	l	a	m
Indeks	0	1	2	3	4	5	6	7	8

Tabel 13. Langkah 7.

Deskripsi	H	a	i		m	a	l	a	m
String					m	a	l	a	m
Indeks	0	1	2	3	4	5	6	7	8

Tabel 14. Langkah 8.

Deskripsi	H	a	i		m	a	l	a	m
String					m	a	l	a	m
Indeks	0	1	2	3	4	5	6	7	8

Tabel 15. Langkah 9.

Deskripsi	H	a	i		m	a	l	a	m
String					m	a	l	a	m
Indeks	0	1	2	3	4	5	6	7	8

Gambar 7. Implementasi Pencarian *String* dengan Algoritma *Knuth Morris Pratt*

Pada gambar 7 menjelaskan bahwa langkah pertama melakukan pencocokan karakter pertama antara *string* dengan deskripsi. Bila belum menemukan kecocokan maka tetap melakukan proses pencocokan. Seperti yang terlihat pada gambar proses pencocokkan *string* dengan deskripsi terjadi sebanyak sembilan langkah pada Algoritma *Knuth Morris Pratt.* Pada langkah kesembilan antara deskripsi dengan *string* mengalami kecocokan di indeks ke-4 maka tidak melakukan pergeseran kembali dan informasi disimpan setelah itu tidak melanjutkan pencocokan deskripsi dengan *string* karena berhenti di indeks ke-8.

Pada algoritma *horspool* tahap pertama pencocokan *string* membuat nilai di tabel *bad match* berdasarkan karakter [17]. *String* malam dicocokan dengan deskripsi Hai malam. Tabel 7 merupakan nilai *bad match string* malam.

Tabel 7. *Bad match*

String	Indeks	Nilai
M	0	4
A	1	3
L	2	2
A	3	1
*	-	5

Pada tabel 7 nilai diperoleh dengan cara berikut.

m = 5-0-1 = 4
a = 5-1-1 = 3
l = 5-2-1 = 2
a = 5-3-1 = 1

Setelah mendapatkan nilai *bad match* langkah selanjutnya adalah melakukan pencocokan *string*. Pada gambar 8 merupakan langkah-langkah dalam penerapan algoritma *Horspool* pada pencarian *string* "malam".

Algoritma *Horspool*

Tabel 17. Langkah 1.

Deskripsi	H a i m a l a m
String	m a l a m
Indeks	0 1 2 3 4

Tabel 18. Langkah 2.

Deskripsi	H a i m a l a m
String	m a l a m
Indeks	0 1 2 3 4

Tabel 19. Langkah 3.

Deskripsi	H a i m a l a m
String	m a l a m
Indeks	0 1 2 3 4

Tabel 20. Langkah 4.

Deskripsi	H a i m a l a m
String	m a l a m
Indeks	0 1 2 3 4

Tabel 21. Langkah 5.

Deskripsi	H a i m a l a m
String	m a l a m
Indeks	0 1 2 3 4

Tabel 22. Langkah 6.

Deskripsi	H a i m a l a m
String	m a l a m
Indeks	0 1 2 3 4

Gambar 8. Implementasi Pencarian *String* dengan Algoritma *Horspool*

Pada gambar 8 di langkah pertama antara *string* dengan deskripsi mengalami kecocokan di indeks ke-4 antara m dan m maka melakukan pergeseran sebanyak 4 karena nilai m = 4 kemudian pergeseran dilakukan pada langkah kedua. Langkah kedua pada gambar 8 antara *string* dengan deskripsi mengalami kecocokan di indeks ke-4 maka tidak melakukan pergeseran kembali dan informasi disimpan setelah itu melanjutkan kembali pencocokan deskripsi dengan *string* pada huruf sebelumnya yaitu huruf a. Proses pencocokkan terus berlangsung sampai pada langkah keenam antara deskripsi dengan *string* mengalami kecocokan secara keseluruhan.

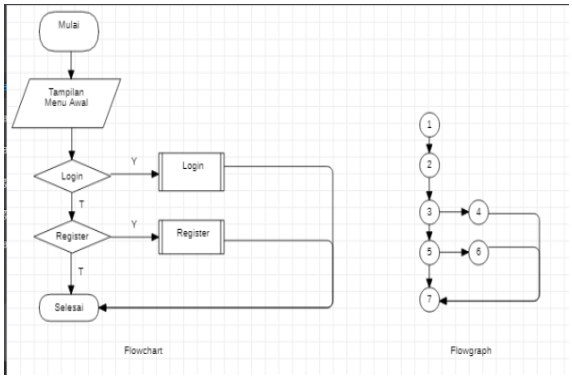
Hasil perbandingan

Pada algoritma *Brute Force* dilakukan pencocokan *string* sebanyak 5 langkah, algoritma *Knuth Morris Pratt* sebanyak 9 langkah, dan algoritma *Horspool* sebanyak 6 langkah. Dalam mempersingkat waktu untuk pencarian *postingan* maka lebih baik menggunakan algoritma *Brute Force*.

Pengujian

Tahapan pengujian dengan menggunakan metode *whitebox* dengan menguji struktur aplikasi, *flowgraph*, dan *flowchart*. Tahapan dengan menggambarkan logika program biasa disebut dengan *Flowchart* sedangkan pengujian alur atau alir program disebut *flowgraph*. Selain membuat *flowchart* dan *flowgraph* melakukan perhitungan yaitu menghitung *region* dengan menghitung *predicate node* seperti *if else flowgraph*, *cyclomatic complexity* menghitung jumlah *path* dari *edge* dan *node*, terakhir *independent path*.

1. *Flowchart* dan *flowgraph* Menu Awal
   Flowchart dan *flowgraph* menu awal dapat dilihat pada gambar 9.

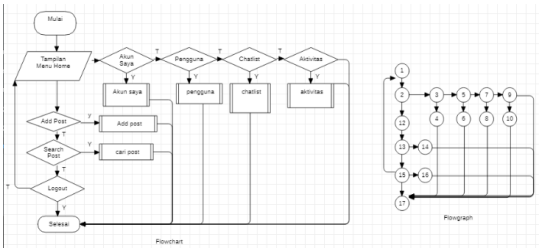


Gambar 9. *Flowgraph* dan *flowchart* menu awal

Tabel 8. *Region*, C*yclomatic Complexity,* dan *Path Flowgraph* Menu Awal

Cyclomatic complexity **nilai *Edge* dan *node***	***Region*** **dengan *predicate Node*(P), P = 2**	***path flowgraph***
E = 8 N = 7 V(G) = E-N+2 = 8-7+2 = 3 Jumlah *path* = 3	V(G) = P + 1 = 2 +1 = 3 *R flowgraph* = 3	*Path* 1 = 1-2-3-4-7 *Path* 2 = 1-2-3-5-6-7 *Path* 3 = 1-2-3-5-7

2. *Flowchart* dan *flowgraph* Menu *Home*
   Flowchart dan *flowgraph* menu *home* dapat dilihat pada gambar 10.

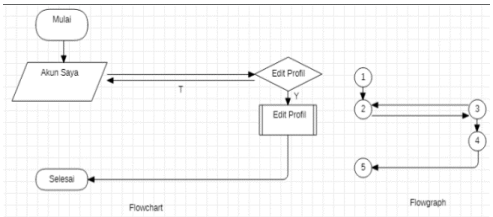


Gambar 10. *Flowgraph* dan *flowchart* menu *home*

Tabel 9. *Cyclomatic complexity, Region*, dan *Path flowgraph* Menu *Home*

Cyclomatic Complexity **nilai *Edge* dan *node***	***Region*** **dengan *predicate Node*(P), P = 7**	***Path flowgraph***
E = 23 N = 17 V(G) = E-N+2 = 23-17+2 = 8 Jumlah *path* = 8	V(G) = P + 1 = 7 +1 = 8 *R flowgraph* = 8	*Path* 1 = 1-2-12-13-15-17 *Path* 2 = 1-2-12-13-15-1-2-12-13-15-1 *Path* 3 = 1-2-12-13-15-16-17 *Path* 4 = 1-2-12-13-14-17 *Path* 5 = 1-2-3-4-17 *Path* 6 = 1-2-3-5-6-17 *Path* 7 = 1-2-3-5-7-8-17 *Path* 8 = 1-2-3-5-7-9-10-17

3. *Flowchart* dan *flowgraph* menu akun saya
   Flowgraph dan *flowchart* menu akun saya dapat dilihat pada gambar 11.



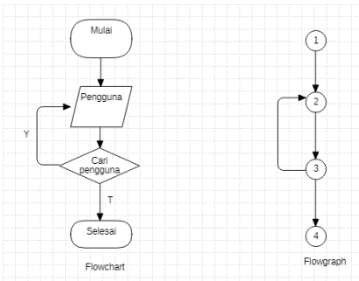
Gambar 11. *Flowgraph* dan *Flowchart* Menu Akun Saya

Tabel 10. *Cyclomatic Complexity, Region,* dan *Path Flowgraph* menu Akun Saya

Cyclomatic Complexity **nilai *Edge* dan *node***	***Region* dengan *predicate Node*(P), P = 1**	***Path flowgraph***
E = 5 N = 5 V(G) = E-N+2 = 5-5+2 = 2 Jumlah *path* = 2	V(G) = P + 1 = 1 +1 = 2 *R flowgraph* = 2	*Path* 1 = 1-2-3-4-5 *Path* 2 = 1-2-3-2-3-4-5

4. *Flowchart* dan *flowgraph* Menu Pengguna

   Flowgraph dan flowchart menu pengguna dapat dilihat pada gambar 12.



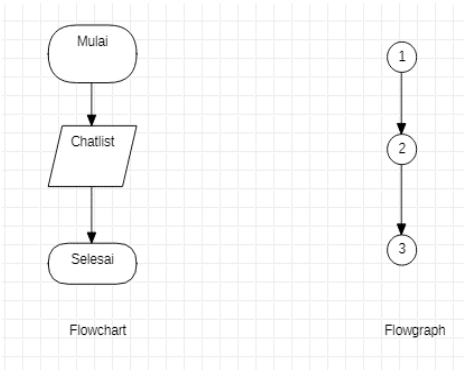
Gambar 12. *Flowgraph* dan *flowchart* menu pengguna

Tabel 11. *Cyclomatic complexity, Region,* dan *Path flowgraph* Menu Pengguna

Cyclomatic Complexity **nilai *Edge* dan *node***	***Region* dengan *predicate Node*(P), P = 1**	***Path flowgraph***
E = 4 N = 4 V(G) = E-N+2 = 4-4+2 = 2 Jumlah *path* = 2	V(G) = P + 1 = 1 +1 = 2 *R flowgraph* = 2	*Path* 1 = 1-2-3-4 *Path* 2 = 1-2-3-2-3-4

5. *Flowchart* dan *flowgraph* Menu *chatlist*

   Flowchart dan *flowgraph* menu *chatlist* dapat dilihat pada gambar 13.

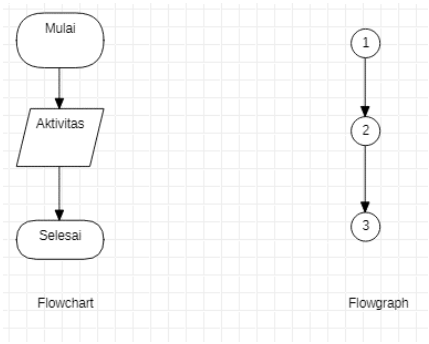


Gambar 13. *Flowgraph* dan *flowchart* menu *chatlist*

Tabel 12. *Cyclomatic Complexity, Region,* dan *Path Flowgraph* Menu *Chatlist*

Cyclomatic Complexity **nilai *Edge* dan *node***	***Region* dengan *predicate Node*(P), P = 0**	***Path flowgraph***
E = 2 N = 3 V(G) = E-N+2 = 2-3+2 = 1 Jumlah *path* = 1	V(G) = P + 1 = 0 +1 = 1 *R flowgraph* = 1	*Path* 1 = 1-2-3

6. *Flowchart* dan *flowgraph* menu aktivitas
   Flowchart dan *flowgraph* menu aktivitas dapat dilihat pada gambar 14.



Gambar 14. *Flowgraph* dan *Flowchart* Menu Aktivitas

Tabel 13. *Cyclomatic Complexity, Region,* dan P*ath Flowgraph* Menu Aktivitas

Cyclomatic Complexity **nilai** ***Edge* dan *node***	***Region* dengan *predicate Node*(P), P = 0**	***Path flowgraph***
E = 2 N = 3 V(G) = E-N+2 = 2-3+2 = 1 Jumlah *path* = 1	V(G) = P + 1 = 0 +1 = 1 *R flowgraph* = 1	*Path* 1 = 1-2-3

Tabel 14. Data Hasil Pengujian *Whitebox*

Flowgraph	***Cyclomatic Complexcity (cc)***	***Region* (R)**	***Independen Path***
Menu awal	3	3	3
Menu *home*	8	8	8
Menu akun saya	2	2	2
Menu pengguna	2	2	2
Menu *chatlist*	1	1	1
Menu aktivitas	1	1	1
Jumlah	17	17	17

Hasil pengujian dengan metode *whitebox* dapat dilihat pada tabel 14. Dengan hasil 3 parameter yang sama yaitu *region, cyclomatic complexcity* , dan *independent path* sebesar 17 maka logika serta alur program sudah sesuai dengan standar ANSI.

Tabel 15. Pengujian Perangkat

Uji	**Cara Pengujian**	**Hasil yang diharapkan**	**Hasil pengujian**
Menu *home*	Menampilkan *postingan*	Dapat melakukan pencarian *post*sing, membuat *post*sing, *logout* dan menekan menu pesan.	Berhasil
Menu akun saya	Tombol akun saya	Menampilkan informasi akun pengguna.	Berhasil
Menu *chatlist*	Tombol *chatlist*	Menampilkan daftar pesan.	Berhasil
Menu pengguna	Tombol pengguna	Menampilkan pengguna yang terdaftar.	Berhasil
Menu aktivitas	Tombol aktivitas	Menampilkan notifikasi komentar dan *like*.	Berhasil

Pengujian menggunakan metode *blackbox* pada data di tabel 15 dapat dikatakan berhasil. Dengan menguji menu *home*, awal, *chatlist*, akun saya, aktivitas dan pengguna.

Pengujian *blackbox* dilakukan dengan tujuan menampilkan aplikasi, cara kegunaannya serta hasilnya sesuai dengan yang diharapkan.

4. SIMPULAN

Aplikasi uSocial merupakan aplikasi media sosial yang dapat saling berkomunikasi satu sama lain seperti media sosial yang sudah ada sebelumnya namun bedanya pada aplikasi ini menerapkan metode *RecyclerView*, *Firebease*, *Volley* serta JSON yang diimplementasikan dalam mengirimkan dan menyimpan data pada saat *push notification*. Selain itu menggunakan penerapan algoritma *Brute Force* dalam pencarian *string* pada navigasi mencari. Pencarian dilakukan berdasarkan panjangnya kalimat dan menghasilkan akurasi sebesar 100% tepat untuk uji coba 500 data. Hasil perbandingan antara algoritma *bruteforce, knuth morris pratt,* dan *horspool* ditemukan bahwa dalam mempersingkat waktu untuk pencarian *postingan* maka lebih baik menggunakan algoritma *Brute Force*. Dalam aplikasi ini pengguna dapat melakukan *posting,* berkomentar, mengirim pesan, dan menyukai *postingan*. Dari pengujian *whitebox* didapatkan hasil yang sama yaitu sebesar 17 pada parameter *region, cyclomatic complexcity, dan independent path* maka alur serta logika program sudah sesuai dengan standar ANSI. Sedangkan hasil pada pengujian *blackbox* fungsi-fungsi yang ada pada aplikasi berhasil menghasilkan *output* yang diinginkan. Dengan menguji menu *home*, awal, aktivitas, pengguna, *chatlist* dan akun saya.

DAFTAR PUSTAKA

[1] Emmadi, Sai Spandhana Reddy, Sirisha Potluri. Android Based Instant *Messaging* Application Using *Firebase*. International Journal of Recent Technology and Engineering (IJRTE)ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019.

[2] Siddik, Mohd, Akmal Nasution. Perancangan Aplikasi *Push notification* Berbasis Android. JURTEKSI (Jurnal Teknologi dan Sistem Informasi)ISSN 2407-1811 (print)Vol. IV No. 2, Jun 2018, hlm. 149 –154.

[3] Sabilla, Shoffi Izza, dkk. Rancang Bangun Aplikasi Perangkat Bergerak Layanan Pemesanan Barang (Studi Kasus "Dinas Kebersihan dan Pertamanan Kota Surabaya"). JURNAL TEKNIK ITS Vol. 5, No. 2, (2016) ISSN: 2337-3539 (2301-9271 Print).

[4] Tahel, Fithry. *Cloud* Server Dalam Pembuatan Aplikasi Fotoblog Realtime Berbasis Android. MAJALAH ILMIAH METHODA Volume 9, Nomor 2, Mei - Agustus2019:64-73; ISSN:2088-9534.

[5] Li, Luqun. RTCA: *Real-time* Communication Application on Android Platform. International Journal of Advance Research in Computer Science and Management Studies Volume 6, Issue 4, April 2018 pg. 74-79.

[6] Rozaq, Afifur, dkk. Pembangunan Aplikasi Brawijaya *Messenger* dengan menggunakan Platform *Firebase* pada Universitas Brawijaya. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X Vol. 2, No. 2, Februari 2018, hlm. 667-673.

[7] Halim, Rony Chandra. Penerapan Algoritma AES dalam Perancangan Aplikasi Media Sosial Berbasis Android. Jurnal ENTER Volume 1, Agustus 2018.

[8] Yapri, Albert Sanada, dkk. Rancang Bangun Aplikasi Mobile IniAtauItu Sebagai Media Tanya Jawab Berbasis

Komunitas. JUISI, Vol. 03, No. 02, Agustus 2017.

[9] Giam, Silvy Krismayanti Winarto, dkk. Aplikasi HIMAINFRA Berbasis Android. Jurnal Infra Vol 6, No 2 2018.

[10] Opoku, Bright, dkk. Development And Implementation Of A New Android Social Media Application[Thesis]. Ghana: Department of Computer Science, Christian Service University College.

[11] N. Fatin, "Pengertian Studi Literatur." [Daring]. Tersedia pada: http://seputarpengertian.blogspot.com/2017/09/pengertian-studi-literatur.html. [Diakses: 16 April 2020].

[12] Rosa, A.S. 2011. Rekayasa Perangkat Lunak. Bandung: Modula.

[13] Lanka, Surekha, dkk. Real Time Scheduler Sistem on Android Mobile App for Academic Institutions. International Journal of Engineering Trends and Technology (IJETT) – Volume 25 Number 1- July 2015.

[14] Jony. 2015. Aplikasi Informasi Akademik Berbasis Android. Jurnal SISFOKOM, Volume 04, Nomor 02, September.

[15] Sumi, A. S., Purnawansyah, P., & Syafie, L. (2018). Analisa Penerapan Algoritma *Brute Force* Dalam Pencocokan *String*. SAKTI (Seminar Ilmu Komputer Dan Teknologi Informasi), 3(2), 88–92.

[16] Nursobah, Pajar Pahrudin, Penerapan Algoritma Pencarian Knuth-Morris-Pratt (KMP) Dalam Sistem Informasi Perpustakaan SMK TI Pratama. SEBATIK 1410-3737, Vol 23 No 1 (2019): Juni 2019.

[17] Puding, Mamta Culkari, dkk. Perbandingan Algoritma Horspool dan Algoritma Raita Pada Aplikasi Istilah Psikologi Berbasis Android. semanTIK, Vol.5, No.1, Jan-Jun 2019, pp. 131-142.

[18] Inspirilo, "2200+ Kosakata Bahasa Inggris dan Artinya yang Paling Sering Digunakan" [Daring]. Tersedia pada: https://inspirilo.com/kosakata-bahasa inggris/ [Diakses: 10 September 2020].