

Pengembangan Aplikasi Sistem Informasi *Smart Register Online* Berbasis Android Menggunakan Algoritma BruteForce

Dimas Dandy Aryarajendra Suprpto¹, Fauziah², Iskandar Fitri³, Nur Hayati⁴
^{1,2,3,4}Program Studi Informatika, Universitas Nasional

email: dimazdandy@gmail.com¹, fauziah@civitas.unas.ac.id²,
iskandarfitri@civitas.unas.ac.id³, nurhayati@civitas.unas.ac.id⁴

(Received: 8 April 2020/ Accepted: 26 April 2020 / Published Online: 20 Juni 2020)

Abstrak

Pada proses pendaftaran mahasiswa baru (PMB) untuk berbagai informasi ditampilkan pada sistem informasi pmb berbasis website. Tetapi kendala dalam penggunaan website adalah calon mahasiswa baru harus sering memantau website untuk memperoleh informasi, bila tidak maka camaba beresiko tertinggal informasi terbaru. Penelitian ini bertujuan untuk mengembangkan sistem informasi registrasi *online* pada pendaftaran mahasiswa baru yang sebelumnya berbasis website dan dikembangkan menjadi aplikasi berbasis Android. Sehingga sangat memudahkan camaba dalam mengakses informasi pmb dan pendaftaran melalui aplikasi Android. Penelitian ini merancang aplikasi sistem informasi pendaftaran berbasis Android menggunakan metode *waterfall*, algoritma brute force, terintegrasi dengan database firebase *realtime* dan cloud firebase *storage*. Hasil pengujian algoritma brute force pada pencarian kata “Teknik Informatika S1” dengan input string “Informatika” dilakukan 8 langkah pencarian. Pada pengujian whitebox dengan menghitung *parameter cyclomatic complexity, region, dan independent path* dari 4 pengujian *whitebox* diperoleh hasil sama yaitu berjumlah 16. Pada hasil pengujian aplikasi dengan metode *blackbox* dinyatakan berhasil dan hasil pengujian sesuai yang diharapkan. Jadi dapat disimpulkan pengembangan aplikasi sistem informasi *smart register online* menggunakan algoritma brute force dikatakan layak digunakan dalam proses pendaftaran mahasiswa baru sesuai dengan pengujian yang telah dilakukan.

Kata kunci: Android, Aplikasi, BruteForce, PMB, Sistem Informasi

Abstract

The process of registering new students (PMB) for sharing information displayed on the website based PMB information system. However, constraints in the usage of the website are prospective students must monitor the website frequently to obtain information. If not accordingly a student is at risk of lagging behind the latest information. The research aims to expand a registration information system on the enrollment of student-based websites and developed into an Android-based application. So it is very easy for students to access PMB information, and registration through the Android application. This study designed an Android-based registration information system application using the waterfall method, brute force algorithm, integrated realtime firebase database and firebase storage cloud. The results of testing the brute force algorithm on the word search "S1 Informatics Engineering" with the input string "Informatics" carried out 8 steps of searching. These results by calculating the parameters of cyclomate complexity, region, and independent path from whitebox tests obtained the same results, the amount of 16. The results of testing blackbox method declared successful and as expected. So it can be concluded the development of an online smart register information apply the brute force algorithm is said to be feasible to implementation in the process of registering new students in accordance with the testing that has been done.

Keywords: Android, Applications, BruteForce, PMB, Information System

PENDAHULUAN

Penerimaan Mahasiswa Baru (PMB) adalah kegiatan yang rutin dilaksanakan oleh Perguruan Tinggi tiap ajaran baru untuk memperoleh mahasiswa baru. Pada proses PMB, berbagai informasi ditampilkan pada sistem informasi PMB. Sistem informasi merupakan sistem yang disusun secara sistematis dan teratur dari jaringan-jaringan aliran informasi yang menghubungkan setiap bagian dari suatu sistem, sehingga memungkinkan diadakannya komunikasi antar bagian atau satuan fungsional (Irawan, Hasna, & Pahlevi, 2016).

Sistem informasi pendaftaran mahasiswa baru atau *smart register online* adalah suatu sistem informasi yang berisi pemberitahuan seputar pmb baik prosedur pendaftaran, biaya kuliah, ujian yang dilakukan, informasi prodi, dan melakukan pendaftaran secara *online*. Biasanya sistem informasi penerimaan mahasiswa baru berupa website. Akan tetapi kendala dalam penggunaan website adalah calon mahasiswa baru harus sering memantau website untuk memperoleh informasi, bila camaba tidak sering memantau website PMB maka mereka beresiko ketinggalan informasi yang *up to date*. Oleh karena itu, untuk menyelesaikan permasalahan diatas dibutuhkan suatu aplikasi sistem informasi PMB berbasis Android dengan bahasa pemrograman Java dan dibuat menggunakan platform Android Studio.

Android adalah sistem operasi mobile yang berkembang saat ini dan berbasis linux kernel yang dirancang untuk perangkat seluler layar sentuh dan komputer tablet. Android yang disebarluaskan secara open source dan menggunakan bahasa pemrograman java (Sitanggang, Loi, & Indra, 2019). Java adalah bahasa pemrograman berorientasi objek dengan elemen arsitektur mendasar dalam suatu program yaitu *class* yang secara kolektif mendefinisikan semua fitur dan fungsi aplikasi (Klausen, 2017). Android Studio adalah lingkungan terpadu Integrated Development Environment (IDE) untuk mengembangkan aplikasi android berdasarkan IntelliJ IDEA yang dibuat oleh Google (Utomo, Budiman, & Triono, 2017).

Selain perkembangan teknologi aplikasi Android yang semakin pesat, aplikasi sistem berbasis Android memudahkan camaba dalam mengakses informasi seputar pmb dan melakukan pendaftaran secara *online* dengan lebih efektif dan efisien. Kelebihan sistem informasi pmb berbasis Android dibanding website adalah camaba tidak perlu mengetikkan alamat dan membuka situs pmb untuk mendapatkan informasi, camaba hanya perlu membuka aplikasi sistem informasi pmb berbasis Android karena lebih efektif dan efisien.

Sejumlah peneliti telah mengembangkan sistem informasi diberbagai bidang menggunakan tool, IDE, dan database yang berbeda beda diantaranya (Amin, Maseleno, Shankar, Perumal, Vidhyavathi, & Lakshmanprabu, 2018; Choi & Park, 2017; Khoiriyah & Manikam, 2019; Kim, Bae, & Choi, 2018; Kouziokas, 2016; Yulistina & Arianti, 2019).

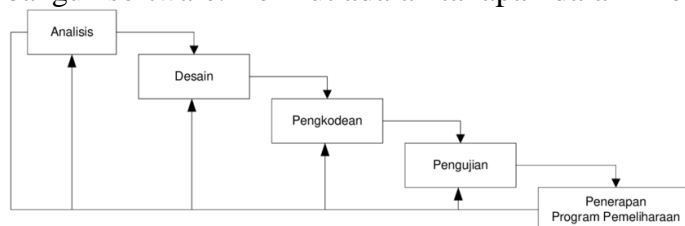
Sementara itu penelitian yang telah dilakukan Pramana, Watrionthos, & Purnama (2019) telah membuat aplikasi sistem informasi pendaftaran mahasiswa baru AMIK Labuhan Batu berbasis android namun baru menerapkan mengenai form pendaftaran yang menggunakan database mysql Kemudian pada hasil penelitian yang dilakukan Yuni (2019) aplikasi penerimaan mahasiswa baru berbasis android menerapkan sistem informasi pendaftaran yang sudah terhubung dengan database firebase dan menyediakan informasi pmb tetapi belum menggunakan algoritma dalam proses pendaftaran.

Berdasarkan pada penelitian sebelumnya, peneliti mengembangkan aplikasi sistem informasi pendaftaran mahasiswa baru secara *online* berbasis Android yang berisikan seluruh informasi PMB mulai dari prosedur pendaftaran, biaya kuliah, lokasi ujian, materi ujian, jenjang akademik (prodi), kelas karyawan, beasiswa, dll. Kemudian melakukan pendaftaran secara *online* melalui aplikasi yang telah terintegrasi dengan database firebase *realtime*, cloud firebase *storage* dan menggunakan metode *waterfall* dalam perancangan aplikasi serta implementasi algoritma brute force dalam proses pendaftaran. Dengan adanya aplikasi sistem informasi PMB berbasis Android, calon mahasiswa baru dapat dipermudah dalam

mendapatkan informasi sehingga tidak ketinggalan informasi penting dan dapat melakukan pendaftaran secara *online* melalui aplikasi Android.

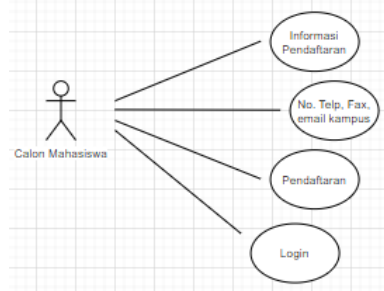
METODE

Penelitian ini menggunakan metode *waterfall* dengan pendekatan sistematis dan berurutan dalam pengembangan aplikasi dan algoritma brute force dalam pencarian string program studi saat melakukan pendaftaran. Metode *waterfall* adalah metode yang dilakukan dalam perancangan suatu sistem dengan secara sistematis dan berurutan. Menurut Pressman (2015:42), metode waterfall adalah metode klasik yang bersifat sistematis, berurutan dalam membangun software. Berikut adalah tahapan dalam metode *waterfall*:



Gambar 1. Metode Waterfall

Langkah awal yaitu analisis terhadap kebutuhan sistem dengan cara pengumpulan data melalui penelitian, observasi, dan studi literatur. Langkah kedua yakni merancang desain tampilan dari aplikasi yang akan dibuat dan membuat use case diagram.



Gambar 2. Use case diagram Rancangan Sistem

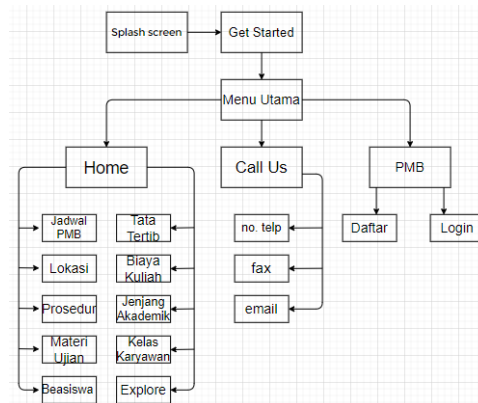
Pada gambar 2 merupakan use case diagram yaitu penjelasan interaksi antara user dengan sistem yang ada yang dipresentasikan dengan langkah sederhana. Langkah ketiga yakni penulisan kode program dengan menggunakan Android Studio bahasa pemrograman java. Langkah keempat adalah pengujian program menggunakan metode *whitebox* dan *blackbox* untuk mendapatkan hasil keberhasilan dalam pengujian aplikasi. Langkah kelima yakni tahap pemeliharaan dilakukan perbaikan pada sistem aplikasi. Algoritma Brute Force adalah algoritma pendekatan secara langsung, sederhana, dan jelas yang digunakan dalam mencocokkan string dengan membandingkan tiap karakter demi karakter. Berikut adalah langkah-langkah algoritma brute force untuk mencocokkan string (Sumi, Purnawansyah, & Syafie, 2018). Pencocokan string dimulai pada awal teks, membandingkan karakter demi karakter dari kiri ke kanan, dan jika pada langkah 1 string belum menemukan kecocokan pada teks akan terus bergeser ke kanan dan berulang ke langkah 2 hingga langkah selanjutnya dan menemukan kecocokan antara teks dan string.

HASIL DAN PEMBAHASAN

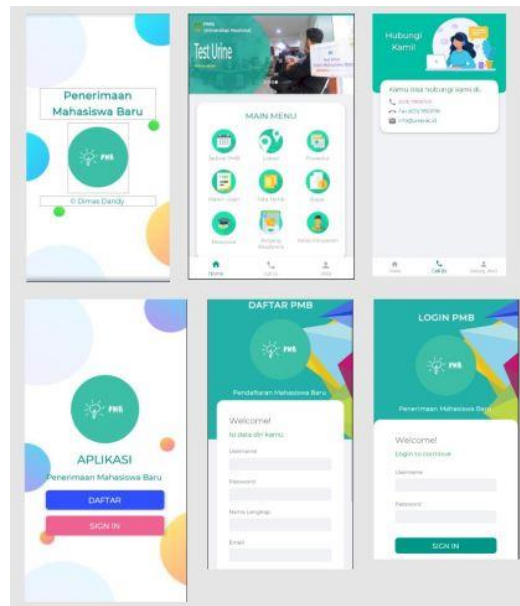
Hasil Penelitian

Aplikasi yang dikembangkan dalam penelitian ini adalah aplikasi *smart register online* berbasis Android menggunakan algoritma brute force. Pengembangan aplikasi dilakukan dengan menggunakan metode pendekatan *Waterfall* dengan langkah-langkah

pelaksanaan penelitian dan pengembangan sebanyak 5 tahapan, yaitu: Analisis sistem, Desain sistem, *Coding*, *Testing*, Pemeliharaan. Hasil pengembangan aplikasi *smart register online* berbasis Android dapat dilihat pada gambar 3 dan 4.



Gambar 3. User Flow

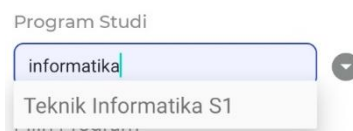


Gambar 4. Tampilan Aplikasi

Pada gambar 3 adalah user flow yaitu langkah yang dilakukan user untuk mencari informasi dan pendaftaran pada aplikasi sistem informasi registrasi *online*. Sementara itu Pada gambar 4 adalah tampilan desain aplikasi yang terdiri dari *splash screen*, main menu Home, menu navigasi *Call us*, menu navigasi PMB, halaman daftar dan login. *Splash screen* adalah halaman ketika membuka aplikasi pertama kali, kemudian masuk main menu Home. Pada Home terdapat menu jadwal PMB, lokasi, prosedur, materi ujian, biaya, beasiswa, jenjang akademik dan kelas karyawan. Kemudian pada menu navigasi *Call us* terdapat kontak PMB berupa nomor telepon, email, dan fax. Pada menu navigasi PMB calon mahasiswa baru dapat melakukan pendaftaran secara *online* melalui aplikasi dan login setelah daftar.

Hasil Uji Implementasi Algoritma Brute Force

Penerapan algoritma *brute force* adalah ketika mengisi form program studi. Algoritma ini dibutuhkan dalam pencarian nama-nama program studi sehingga camaba tidak perlu mengetikkan seluruh teks program studi yang dicari. Pada form ini juga diberikan *option selection* menggunakan *dropdown*.



Gambar 5. Form Pengisian Program Studi

Pada gambar 5 merupakan form pengisian program studi. Disini user akan mencari jurusan teks “Teknik Informatika S1” dengan *string* “informatika”. Berikut adalah penerapan algoritma *brute force* pada kasus diatas.

1. Penelusuran Langkah 1

Tabel 1. Langkah 1

Langkah 1	
Prodi	T e k n I k I n f o r m a t i k a S 1
String	I n f o R m a t i k a

Pada tabel 1 bandingkan huruf pertama string input dengan string pertama prodi, terjadi proses pencocokan string atau huruf yang sama. Bila belum terjadi kecocokan antra string prodi dan string input, maka proses pencocokan akan terus berlangsung.

2. Penelusuran Langkah 2

Tabel 2. Langkah 2

Langkah 2	
Prodi	T e k n I k I n f o r m a t i k a S 1
String	■ I n f O r m a t i k a

Pada tabel 2 masih sama seperti tabel 1, belum bertemu string yang cocok sehingga proses masih berlanjut.

3. Penelusuran Langkah 3

Tabel 3. Langkah 3

Langkah 3	
Prodi	T e k n I k I n f o r m a t i k a S 1
String	■ ■ I n F o r m a t i k a

Pada tabel 3 belum bertemu string yang cocok sehingga proses masih berlanjut. Hingga string prodi dan string input cocok.

4. Penelusuran Langkah 4

Tabel 4. Langkah 4

Langkah 4	
Prodi	T e k n I k I n f o r m a t i k a S 1
String	■ ■ ■ I N f o r m a t i k a

Pada tabel 4 string belum cocok, tetap melakukan pergeseran ke kanan.

5. Penelusuran Langkah 5

Tabel 5. Langkah 5

Langkah 5	
Prodi	T e k n i k I n f o r m a t i k a S 1
String	■ ■ ■ ■ I n f o r m a t i k a

Pada tabel 5 string belum cocok, tetap melakukan pergeseran ke kanan.

6. Penelusuran Langkah 6

Tabel 6. Langkah 6

Langkah 6	
Prodi	T e k n i k I n f o r m a t i k a S 1
String	■ ■ ■ ■ ■ I n f o r m a t i k a

Pada tabel 6 string belum cocok, tetap melakukan pergeseran ke kanan.

7. Penelusuran Langkah 7

Tabel 7. Langkah 7

Langkah 7			
Prodi	T e k n i k	I n f o r m a t i k a	S l
String		I n f o r m a t i k a	

Pada tabel 7 string belum cocok, tetap melakukan pergeseran ke kanan.

8. Penelusuran Langkah 8

Tabel 8. Langkah 8

Langkah 8			
Prodi	T e k n I k	I n f o r m a t i k a	S l
String		I n f o r m a t i k a	

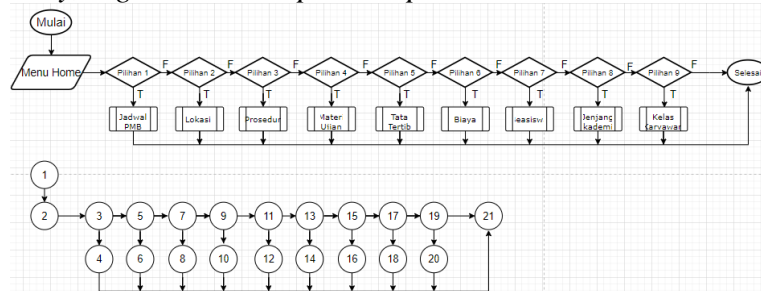
Pada tabel 8 ditemukan bahwa string input sama dengan string prodi. Maka tidak lagi dilakukan pergeseran atau pencarian string kembali. Setelah melakukan pengujian hasil implementasi algoritma brute force, selanjut nya adalah pengujian aplikasi menggunakan metode *whitebox* dan *blackbox*.

Hasil Pengujian Metode Whitebox pada Aplikasi

Metode *whitebox* adalah pengujian software yang paling terstruktur dengan melihat kode dan struktur data. Pada pengujian aplikasi menggunakan *whitebox* terlebih dahulu membuat *flowchart* dan *flowgraph*. *Flowchart* digunakan pada tahapan perancangan untuk menggambarkan logika dari program sedangkan *flowgraph* digunakan pada tahapan pengujian yang berfokus pada penggambaran aliran dari sebuah program. Pada perhitungan *whitebox* menguji dengan menghitung 3 parameter yaitu *cyclomatic complexity*, *region*, dan *independent path*. *Cyclomatic complexity* adalah menghitung jumlah path dari *edge* dan *node* yang ada pada *flowgraph*. Kemudian menghitung *region* berdasarkan *predicate node* yaitu jumlah *if else* pada *flowgraph*. Langkah terakhir adalah menyusun path pada *flowgraph* berdasarkan jumlah path yang dihitung pada *cyclomatic complexity*.

1. *Flowchart* dan *Flowgraph* Menu Home

Pada gambar 6 adalah *flowchart* (logika program) dan *flowgraph* (aliran program) pada menu home. Setelah menganalisis *flowchart* dan *flowgraph* adalah menghitung *cyclomatic complexity*, *region*, dan *independent path*.



Gambar 6. *Flowchart* dan *flowgraph* Menu Home

Tabel 9. Perhitungan *Cyclomatic complexity*, *Region*, dan Path pada *Flowgraph* Menu Home

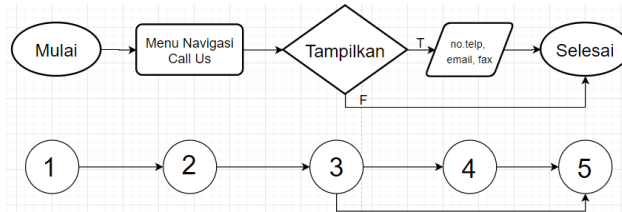
Menghitung <i>Cyclomatic Complexity</i> dari <i>Edge</i> dan <i>Node</i>	Menghitung <i>Region</i> berdasarkan <i>Predicate Node (P)</i> , dimana $P = 9$.
$E = 29$	$V(G) = P + 1$
$N = 21$	$= 9 + 1$
$V(G) = E - N + 2$	$= 10$
$= 29 - 21 + 2 = 10$	Jadi <i>Region (R)</i> pada <i>flowgraph</i> = 10
Jadi, jumlah <i>Path</i> = 10 <i>Path</i>	

Path pada flowgraph di atas

- Path 1 = 1-2-3-4-21
- Path 2 = 1-2-3-5-6-21
- Path 3 = 1-2-3-5-7-8-21
- Path 4 = 1-2-3-5-7-9-10-21
- Path 5 = 1-2-3-5-7-9-11-12-21
- Path 6 = 1-2-3-5-7-9-11-13-14-21
- Path 7 = 1-2-3-5-7-9-11-13-15-16-21
- Path 8 = 1-2-3-5-7-9-11-13-15-17-18-21
- Path 9 = 1-2-3-5-7-9-11-13-15-17-19-20-21
- Path 10 = 1-2-3-5-7-9-11-13-15-17-19-21

2. Flowchart dan Flowgraph Menu Call us

Pada gambar 7 adalah flowchart (logika program) dan flowgraph (aliran program) pada menu call us. Setelah itu menghitung cyclomatic complexity, predicate node (region), dan independent path.



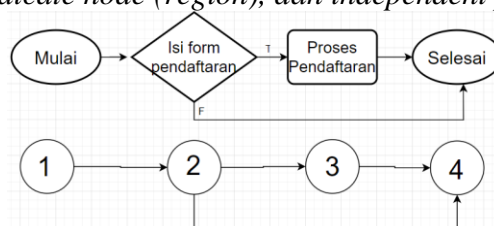
Gambar 7. Flowchart dan flowgraph Menu Call us

Tabel 10. Perhitungan Cyclomatic complexity, Region, dan Path pada Flowgraph Menu Call us

Menghitung Cyclomatic Complexity dari Edge dan Node	Menghitung Region berdasarkan Predicate Node (P), dimana P = 1.	Path pada flowgraph di atas
$E = 5$	$V(G) = P + 1$	Path 1 = 1-2-3-4-5
$N = 5$	$= 1 + 1$	Path 2 = 1-2-3-5
$V(G) = E - N + 2$	$= 2$	
$= 5 - 5 + 2$	Jadi Region (R) pada flowgraph = 2	
$= 2$		
Jadi, jumlah Path = 2 Path		

3. Flowchart dan Flowgraph Menu Daftar

Pada gambar 8 adalah flowchart (logika program) dan flowgraph (aliran program) pada menu daftar. Setelah menganalisis flowchart dan flowgraph adalah menghitung cyclomatic complexity, predicate node (region), dan independent path.



Gambar 8. Flowchart dan flowgraph Menu Daftar

Tabel 11. Perhitungan Cyclomatic complexity, Region, dan Path pada Flowgraph Menu Daftar

Menghitung Cyclomatic Complexity dari Edge dan Node	Menghitung berdasarkan Predicate Node (P), dimana P = 1.	Path pada flowgraph di atas
$E = 4$	$V(G) = P + 1$	Path 1 = 1-2-3-4
$N = 4$	$= 1 + 1$	Path 2 = 1-2-4
$V(G) = E - N + 2$	$= 2$	

$$= 4 - 4 + 2$$

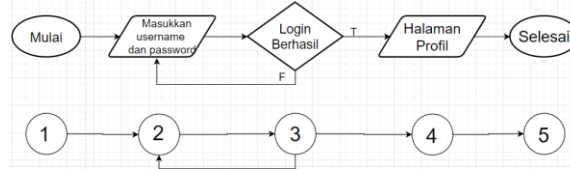
$$= 2$$

Jadi *Region (R)* pada *flowgraph* = 2

Jadi, jumlah Path = 2 Path

4. Flowchart dan Flowgraph Menu Login

Pada gambar 9 adalah *flowchart* (logika program) dan *flowgraph* (aliran program) pada menu login. Setelah menganalisis *flowchart* dan *flowgraph* adalah menghitung *cyclomatic complexity*, *predicate node* (*region*), dan *independent path*.



Gambar 9. Flowchart dan flowgraph Menu Login

Tabel 12. Perhitungan *Cyclomatic complexity*, *Region*, dan Path pada *Flowgraph* Menu Login

Menghitung <i>Cyclomatic Complexity</i> dari Edge dan Node	Menghitung berdasarkan Predicate Node (P), dimana P = 1.	Path pada <i>flowgraph</i> di atas
E = 5	V(G) = P + 1	Path 1 = 1-2-3-2-3-4-5
N = 5	= 1 + 1	Path 2 = 1-2-3-4-5
V(G) = E - N + 2	= 2	
= 5 - 5 + 2	Jadi <i>Region (R)</i> pada	
= 2	<i>flowgraph</i> = 2	
Jadi, jumlah Path = 2 Path		

Tabel 13. Hasil pengujian aplikasi dengan metode *whitebox*

Flowgraph	Cyclomatic Complex city (cc)	Region (R)	Independen Path
Menu Home	10	10	10
Menu Call us	2	2	2
Menu Daftar	2	2	2
Menu Login	2	2	2
Jumlah	16	16	16

Pada tabel 13 adalah hasil pengujian aplikasi dengan metode *whitebox* secara keseluruhan. Dengan mendapatkan jumlah *cyclomatic complexity*, *region*, dan *independent path* yang sama yaitu 16, artinya logika dan alur program yang dibuat sudah sesuai yang diharapkan dan tidak perlu melakukan *compile* ulang.

Hasil Pengujian Metode *Blackbox* pada Aplikasi

Metode *blackbox* adalah pengujian terhadap fungsi operasi dan kegunaan aplikasi apakah pengujian berhasil dan hasil yang didapat sudah sesuai yang diharapkan atau belum. Pada tabel 14 adalah hasil pengujian aplikasi dengan metode *blackbox*.

Tabel 14. Hasil pengujian aplikasi dengan metode *blackbox*

Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil pengujian
Menu Home	Masuk ke halaman utama	Menampilkan menu jadwal, lokasi, prosedur, materi ujian, tata tertib, biaya, beasiswa, jenjang akademik, kelas karyawan, serta navigasi	Berhasil

		<i>call us</i> dan PMB	
Menu Call Us	Menekan navigasi <i>call us</i>	Menampilkan informasi kontak PMB	Berhasil
Menu Daftar	Menekan tombol pendaftaran pada navigasi PMB dan menginput data form	Menampilkan formulir pendaftaran dan data berhasil diinput serta tersimpan dalam database	Berhasil
Menu Sign In	Login dengan username password yang telah didaftarkan	Menampilkan halaman <i>sign in</i> dan berhasil masuk ke menu data profil	Berhasil

Pembahasan

Aplikasi sistem informasi *smart register online* berbasis Android dikembangkan menggunakan metode *waterfall* memiliki halaman menu utama *Home*. Pada halaman menu utama memiliki 3 tombol *bottom* navigasi menu yaitu *home*, *call us*, dan *pmb*. Pada menu utama atau *Home* terdiri dari 9 main menu yang terdiri dari jadwal (berisi jadwal pendaftaran), lokasi ujian, prosedur *pmb*, materi ujian, tata tertib, biaya kuliah, beasiswa, jenjang akademik, dan kelas karyawan. Selanjutnya pada *bottom* navigasi menu *call us* berisi nomor telepon, fax, dan email *customer service* *pmb*. Bila calon mahasiswa baru masih belum jelas mengenai informasi yang disampaikan pada aplikasi atau ada masalah ketika pendaftaran online camaba dapat menghubungi kontak yang ada. Kemudian navigasi menu PMB berisi 2 menu yaitu *daftar* dan *sign in*. Pada menu *daftar* adalah proses registrasi pendaftaran camaba dengan mengisi formulir. Setelah melakukan proses pendaftaran camaba dapat melakukan *sign in* atau login untuk melihat hasil registrasi pendaftaran yang telah berhasil. Selanjutnya adalah pembahasan tentang implementasi algoritma brute force. Pada pengimplementasian algoritma brute force pada proses pengisian form program studi dengan pencarian kata “Teknik Informatika S1” menggunakan string “Informatika” didapatkan dengan melakukan 8 langkah pencarian. Pencarian kata menggunakan brute force dimulai dari kiri dan bergeser terus ke kanan hingga string yang diinputkan cocok dengan kata yang dicari. Dengan penggunaan algoritma brute force akan lebih memudahkan dalam pencarian daftar prodi tanpa harus mengetik seluruh kata. Kemudian dilakukan pengujian aplikasi menggunakan metode *whitebox* dan *blackbox*.

Hasil Pengujian menggunakan metode *whitebox* dilakukan pembuatan *flowchat*, *flowgraph*, serta 4 pengujian terhadap menu utama *Home*, menu *call us*, menu *daftar*, dan menu *sign in* dengan menghitung 3 parameter yaitu *cyclomatic complexity*, *region*, dan *independent path*. Adapun pengujian pada menu utama *Home* didapatkan hasil pada *cyclomatic complexity* yaitu 10 *path*, 10 *region*, dan 10 *independent path*. Kemudian pengujian pada menu *call us*, menu *daftar*, dan menu *sign in* didapatkan hasil yang sama yakni dengan hasil *cyclomatic complexity* yaitu 2 *path*, 2 *region*, dan 2 *independent path*. Maka diperoleh hasil yang berjumlah sama yaitu 16 pada ketiga parameter yang diuji.

Pengujian aplikasi menggunakan metode *blackbox* dilakukan bentuk pengujian terhadap hasil yang diharapkan, dan dari pengujian yang telah dilakukan dinyatakan berhasil. Pengujian dilakukan pada menu utama *home*, menu *call us*, menu *daftar*, dan menu *sign in*.

SIMPULAN

Pengembangan Aplikasi sistem informasi smart register online berbasis Android menggunakan metode waterfall dalam pengembangan aplikasi yaitu dengan tahapan analisis sistem, desain sistem, coding, testing program, dan pemeliharaan. Kemudian dengan implementasi algoritma brute force membantu dalam pencarian string program studi yaitu kata “Teknik Informatika S1” dengan string “Informatika” diperoleh dengan 8 langkah pencarian. Hasil pengujian *whitebox* dengan menghitung *parameter cyclomatic complexity, region, dan independent path* dari 4 pengujian *whitebox* diperoleh hasil sama yaitu berjumlah 16. Pada hasil pengujian aplikasi dengan metode *blackbox* dinyatakan berhasil dan hasil pengujian sesuai yang diharapkan.

REFERENSI

- Amin, M. M., Maselena, A., Shankar, K., Perumal, E., Vidhyavathi, R. M., & Lakshmanaprabu, S. K. (2018). Active Database System Approach and Rule Based in the Development of Academic Information System. *International Journal of Engineering & Technology*, 7(2.26), 95-101.
- Choi, Y., & Park, D. H. (2017). Development of Youke mining system with Youke's travel demand and insight based on web search traffic information. *Journal of Intelligence and Information Systems*, 23(3), 155-175.
- Irawan, A., Hasna, A., & Pahlevi, R. (2016). Sistem Informasi Perdagangan Pada PT Yoltan Sari Menggunakan PHP Berbasis Web. *Jurnal Positif*. 1(2), 8–15.
- Khoiriyah, S., & Manikam, R. M. (2019). Analisa dan Perancangan Sistem Perhitungan Insentif Marketing Trade Menggunakan Sistem Remunerasi. *EDUMATIC: Jurnal Pendidikan Informatika*, 3(2), 99-108.
- Kim, S. J., Bae, Y. M., & Choi, Y. J. (2018). Design and Implementation of Real-time Augmented Reality Building Information System Combined with 3D Map. *Journal of The Korea Computer Graphics Society*, 24(4), 39-54.
- Klausen, P. (2017). *Java 1: Basic syntax and semantics: Software Development*. Diambil dari <https://bit.ly/2AvJgMV>.
- Kouziokas, G. N. (2016). Geospatial based information system development in public administration for sustainable development and planning in urban environment. *European Journal of Sustainable Development*, 5(4), 347-347.
- Pramana, A., Watrianthos, R., & Purnama, I. (2019). Sistem Informasi Pendaftaran Mahasiswa Baru Berbasis Android. *Jurnal Informatika Upgris*, 5(2), 121–125.
- Pressman, R.S. (2015). *Rekayasa Perangkat Lunak: Pendekatan Praktisi Buku I*. Yogyakarta: Andi.
- Sitanggang, A. L., Loi, M. H., & Indra, E. (2019). Rancang Bangun Aplikasi Sistem Ujian Masuk SMA Berbasis Android (Studi Kasus SMA Amir Hamzah Medan). *Jurnal Sistem Informasi Ilmu Komputer Prima (JUSIKOM PRIMA)*, 2(2), 8-16.
- 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.
- Utomo, R. P., Budiman, A., & Triono, J. (2017). Aplikasi Ujian Online Masuk Universitas Merdeka Madiun Berbasis Android. *Jurnal Pilar Teknologi*, 2(1), 29–39.
- Yulistina, D., & Arianti, B. D. D. (2019). E-Katalog Sebagai Sistem Informasi Pemasaran Kopi Sapat Berbasis Web. *EDUMATIC: Jurnal Pendidikan Informatika*, 3(2), 45-52.
- Yuni, P. (2019). Aplikasi Berbasis Android Penerimaan Mahasiswa Baru pada Universitas PGRI Madiun. *SENATIC (Seminar Nasional Teknologi Informasi Dan Komunikasi)*, 2(1), 125–127.