



Rancang Bangun MyAKAD Apps Berbasis Android Menggunakan Algoritma Sequential Searching

Wahyu Mahatma Kurniawan*, Fauziah, Aris Gunaryati

Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional, Jakarta, Indonesia

Email: ^{1,*}wahyumahatma.k@gmail.com, ²fauziah@civitas.unas.ac.id, ³aris.gunaryati@civitas.unas.ac.id

Email Penulis Korespondensi: wahyumahatma.k@gmail.com

Abstrak—Eksistensi dunia digital telah memberikan dampak baru untuk segala aktivitas manusia. Tak heran, jika saat ini semua beralih ke dunia digital karena di satu sisi dapat membuat segala sesuatunya menjadi lebih efisien dan praktis. Tujuan penelitian ini, penulis ingin mengembangkan sebuah aplikasi administrasi data mahasiswa yang akan diterapkan di salah satu Universitas dengan sistem platform berbasis android. Dimana rancangan ini dibuat untuk menyempurnakan keterbatasan aplikasi yang sebelumnya telah dikembangkan yaitu web, Keterbatasan aplikasi sebelumnya di Universitas tersebut, saat pengguna (user) mengakses ke menu akademik online di aplikasi web tersebut, diharuskan menonaktifkan pop-up terlebih dahulu, itu yang dirasa tidak efisien. Berdasarkan permasalahan yang ada, penulis ingin merancang sebuah aplikasi yang akan dikembangkan pada platform android dengan menggunakan algoritma sequential searching pada setiap fungsi program pada aplikasi. Metode pengembangan sistemnya sendiri menggunakan metodologi Waterfall. Metode ini memiliki tahapan yaitu: Analysis, Design, Coding dan Testing. Teknik pengumpulan datanya didapat dari hasil observasi (pengamatan) dan studi pustaka. Database sebagai media penyimpanan data adalah MySQL. Untuk bahasa pemrograman PHP dan Java digunakan sebagai pembuatan interface aplikasi. Hasil pengujian aplikasi direpresentasikan pada White box testing dan menunjukkan nilai valid dari hasil pengujian berjumlah 6 pada total keputusan menunjukkan algoritma yang digunakan cukup baik dan memenuhi standar.

Kata Kunci: Android, Algoritma Sequential Searching, Java, Waterfall dan White Box Testing

Abstract—The existence of the digital world has given a new impact to all human activities. Not surprisingly, if now all switch to the digital world because on the one hand can make things more efficient and practical. The purpose of this study, the author wants to develop a student data administration application that will be applied at one of the universities with an Android-based platform system. Where this design is made to perfect the limitations of applications that have previously been developed, namely the web, limitations of previous applications at the University, when users (users) access to the online academic menu in the web application, are required to deactivate pop-ups first, that is felt to be inefficient. Based on the existing problems, the author wants to design an application that will be developed on the Android platform by using a sequential searching algorithm for each program function in the application. The system development method itself uses the Waterfall methodology. This method has stages namely: Analysis, Design, Coding and Testing Data collection techniques are obtained from the results of observations (observations) and literature studies. Database as a data storage medium is MySQL. For the PHP and Java programming languages used as an application interface, the application test results are represented in the White box testing and show a valid value of the test results amounting to 6 in the total decision shows the algorithm used is quite good and meets the standard.

Keywords: Android, Algoritma Sequential Searching, Java, Waterfall dan White Box Testing

1. PENDAHULUAN

Sistem pengelolaan data administrasi akademik lembaga pendidikan memang masih belum bisa dikatakan baik, dan perlu adanya pengembangan desain penyimpanan data akademik yang *up to date* untuk mendukung berjalannya sistem administrasi akademik[1]. Permasalahan yang masih ada pada sistem pengelolaan akademik sebelumnya yaitu untuk mengakses menu layanan akademik, user harus memblokir *pop-up* terlebih dahulu. Dengan pembaharuan sistem yang dilakukan pada aplikasi data mahasiswa akademik ini diharapkan dapat meningkatkan performa sistem semakin baik dan lebih efisien[2]. Penelitian di fokuskan pada ruang lingkup akademik. Bidang pendidikan yang bertujuan mempelajari segala sesuatu hal tentang ilmu pengetahuan maupun pembelajaran baik secara teori dan praktik guna meningkatkan pengetahuan dalam hal disiplin ilmu, akademik juga melakukan aktivitas lainnya secara rutin seperti administrasi akademik untuk menunjang pelayanan[5].

Beberapa peneliti pernah melakukan penelitian tentang sistem akademik ini. Penelitian oleh Lucy Sri Musmini dan Ni Ketut Kertiasih merancang sistem akademik berbasis *website* menggunakan bahasa pemrograman *HTML* dan *PHP* serta menggunakan *database SQL* pada *MySQL*[1]. Penelitian serupa juga pernah dilakukan oleh W.F. Prasetyo yang merancang sistem akademik berbasis *android* menggunakan sistem *sharedpreference* dan *database MySQL*[2]. Mohammad Rizal dan Fitri Latifah juga pernah melakukan penelitian menggunakan algoritma *sequential search* yang berbasis *android*[3]. Penelitian oleh Ginanjar Wiro Sasmito membahas mengenai penerapan metode pengembangan *waterfall* pada desain sistem informasi berbasis *website* yang menggunakan bahasa pemrograman *PHP* dan *Codeigniter* serta *database MySQL*[4]. Penelitian yang dilakukan Henry Setiana dan Seng Hansun merancang sistem akademik dengan menerapkan metode *game* didalamnya[5]. Penelitian oleh Dedy Setiawan membangun sistem akademik menggunakan bahasa pemrograman *Java* dan *database MySQL*. Metode *waterfall* digunakan untuk analisis dan desain pada sistem[6]. Penelitian oleh Divya R. Sharma dan kawan-kawan merancang sistem akademik berbasis *website* aplikasi dan *android* aplikasi. Sistem *website* dirancang menggunakan *server .Net* pada Visual Studio 2015 dan menggunakan bahasa pemrograman *C#*. Untuk *android* dibangun menggunakan bahasa pemrograman *Java* pada Android Studio[7]. Anisya Sonita dan Mayang Sari juga

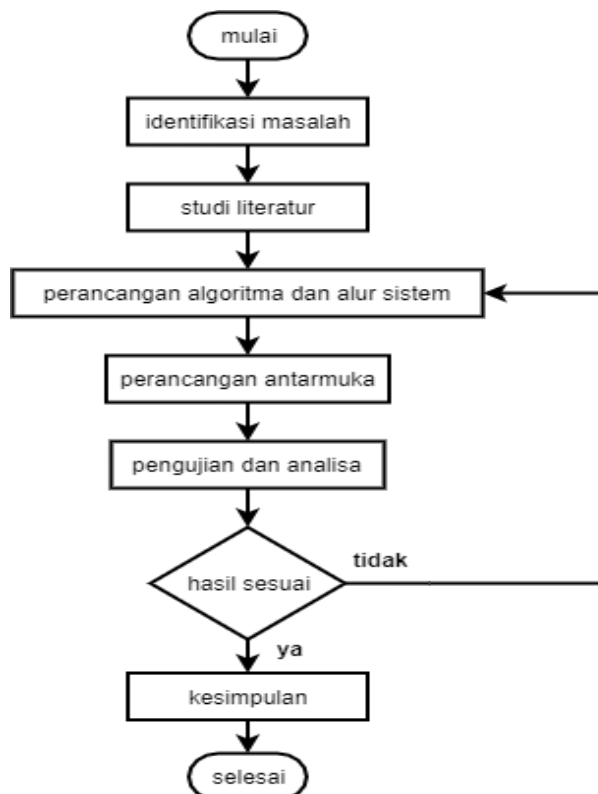


pernah melakukan penelitian mengenai penerapan algoritma *sequential searching* pada suatu sistem berbasis *web* aplikasi menggunakan bahasa pemrograman *Java*[8]. Penelitian oleh Akanksha Verma melakukan penelitian tentang penerapan teknik pengujian *white box* yang dilakukan pada suatu sistem aplikasi[9]. Penelitian yang dilakukan oleh Dea Yuzistin yang merancang sistem akademik siswa berbasis *website* menggunakan bahasa pemrograman *PHP* dan *database MySQL*[10]. Penelitian selanjutnya dilakukan oleh Indra Irawan yang merancang sistem akademik menggunakan jaringan *LAN* berbasis *TCP/IP*, menggunakan bahasa pemrograman Visual Basic 6.0 dan *database Ms. SQL Server 2000*[11]. Penelitian oleh Khaerul Anam dan Asep Taufik Muharram yang merancang sistem akademik berbasis *web* dengan metode *waterfall*, bahasa pemrograman *PHP* dan *database MySQL*[12]. Penelitian yang dilakukan oleh Muhamad Ibrahim dan Arwan Hermawan yang merancang sistem akademik dengan dua sisi, yaitu pertama sisi *client* berupa aplikasi pada platform *android (smartphone)* menggunakan *SDK* yang merupakan *framework Java Mobile*. Kedua adalah sisi *server* yang berbasis *website* yang dirancang dengan bahasa pemrograman *PHP* dan *database MySQL*[13].

Berdasarkan penelitian yang sebelumnya dilakukan oleh para peneliti yang membahas tentang sistem akademik, dapat diuraikan permasalahan dalam pengelolaan sistem akademik. Diantaranya adalah perancangan sistem akademik sebelumnya masih kurang baik, desain penyimpanan data belum up to date, dan tidak efisiensi waktu karena user harus memblokir pop-up untuk mengakses menu layanan akademik. Maka dari itu dirancanglah sistem akademik berbasis android aplikasi bernama MyAKAD. Aplikasi MyAKAD memiliki opsi berikut di halaman awal seperti: *login user* dan *password*. Untuk *user* yang sudah melakukan registrasi akun dapat mengakses berbagai jenis informasi yang ada pada aplikasi [7]. Rancangan aplikasi menggunakan standar algoritma yang telah digunakan sebelumnya oleh para pengembang perangkat lunak. algoritma *sequential searching* yakni menjelaskan setiap fungsi yang ada pada program dan menjelaskan tiap baris pada logika pemrogramannya berurutan dan melakukan *testing* [3]. Berdasarkan kondisi yang telah dibahas, penulis tertarik mempelajari masalah tersebut lebih jauh dengan merancang sebuah aplikasi pengelolaan data administrasi. Memfokuskan penelitian dengan mengumpulkan beberapa analisa pada sistem, pengumpulan data secara kuantitatif serta mendesain rancangan aplikasi lalu melakukan *testing* pada *white box* untuk menyempurnakan sistem. Tujuan penelitian ini untuk menguraikan hasil penelitian aplikasi administrasi akademik yang sebelumnya dengan sistem yang akan dirancang oleh penulis serta menampilkan hasil pengujian perangkat lunak pada rancangan sistem dengan teknik *white box testing*.

2. METODOLOGI PENELITIAN

Berikut gambaran metode penelitian yang dilakukan dalam penelitian ini.



Gambar 1. Flowchart Metode Penelitian

**a. Identifikasi Masalah**

Tahap ini melakukan identifikasi permasalahan pada system pengelolaan akademik universitas yang masih belum baik dan perlu pengembangan aplikasi. Selain itu untuk mengakses menu layanan akademik, user harus memblokir *pop-up* terlebih dahulu yang sangat merepotkan dan membuang banyak waktu.

b. Studi Literatur

Dilakukan dengan cara mempelajari dan mengumpulkan informasi melalui buku-buku literatur, jurnal akreditasi ISSN ataupun jurnal lainnya sebagai sumber referensi pustaka. Selain itu juga untuk menentukan perangkat software dan hardware yang diperlukan serta kebutuhan sistem yang efektif dan efisien. Sehingga dapat mengurangi kesalahan dan untuk memaksimalkan hasil dari penelitian.

c. Pengujian dan Analisa

Pengujian yang dilakukan adalah melakukan ujicoba terhadap proses kerja algoritma sequential searching dan aplikasi MyAKAD yang dibuat menggunakan teknik white box. Semua data hasil ujicoba dicatat secara lengkap untuk selanjutnya dianalisa. Proses analisis dilakukan dengan menentukan apakah hasil yang didapat sesuai dengan yang diharapkan atau tidak. Jika hasil belum sesuai, maka penelitian akan diulangi dari tahap perancangan algoritma dan alur sistem.

d. Penulisan Kesimpulan

Penulisan kesimpulan dilakukan jika hasil ujicoba dari alat dan aplikasi sudah sesuai dengan yang diharapkan. Isi kesimpulan akan menjadi laporan dari penelitian yang dilakukan.

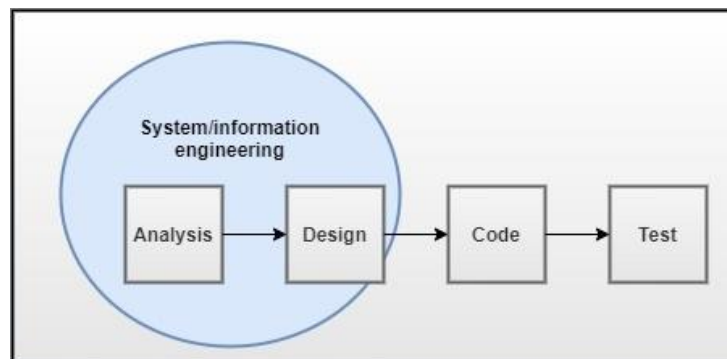
e. Spesifikasi Perangkat Lunak dan Perangkat Keras**Tabel 1.** Perangkat Lunak dan Perangkat Keras Yang Dibutuhkan

Spesifikasi Perangkat Utama Yang Digunakan	Spesifikasi Perangkat Pendukung
Xampp (<i>Web server</i>), Google Chrome	Sistem Operasi <i>Android</i> 6.0
Dia dan EdrawMax (<i>Diagram design</i>)	RAM 3 GB, Internal 32 GB
Android Studio versi 6.0	Octa Core A53 2.0GHz
OS Windows 10 (64-bit), RAM 16 GB, HDD 1 TB	Qualcomm MSM8953 Snapdragon 625
Visual Studio Code	GPU : Adreno 506

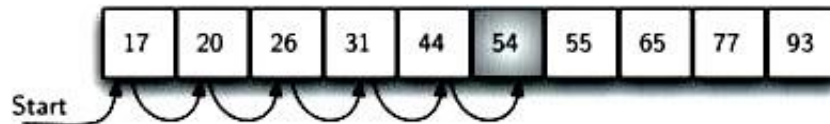
Tahapan ini dilakukan dalam memenuhi spesifikasi kebutuhan perangkat lunak dan perangkat keras pada sistem yang ingin dibangun. Segala kebutuhan dalam membangun sebuah aplikasi berbasis *platform android* ini melibatkan segala kebutuhan perangkat yang akan digunakan dalam perancangan *interface* aplikasi[4].

3. HASIL DAN PEMBAHASAN

Proses pertama pada tahap ini adalah pemilihan metode pengembangan aplikasi, yaitu Metode *Waterfall*. Kedua adalah perancangan algoritma menggunakan Algoritma *Sequential Searching*. Ketiga adalah penjelasan alur sistem dalam bentuk *flowchart*.

**Gambar 2.** Tahap-tahap Metode *Waterfall*

Tahap pertama pada metode pengembangan *waterfall* adalah *analysis*, yaitu pengumpulan data dengan metode kuantitatif. Selanjutnya adalah *design*, adalah proses mengalokasikan kebutuhan-kebutuhan untuk membangun aplikasi baik perangkat keras maupun perangkat lunak dalam membentuk arsitektur aplikasi secara keseluruhan. Kemudian tahap *code*, tahap merancang sistem dengan Bahasa pemrograman yang dipilih. Terakhir adalah tahap *test*, melakukan proses uji coba aplikasi dengan teknik ujicoba yang ada.



Gambar 3. Algoritma *Sequential Searching*

Pencarian merupakan proses yang fundamental dalam pengolahan data. *Sequential searching* ialah proses membandingkan setiap elemen array satu persatu secara berurut[8]. Urutan-urutan fungsi pada Algoritma *Sequential Searching* adalah sebagai berikut ini :

$i < 0$

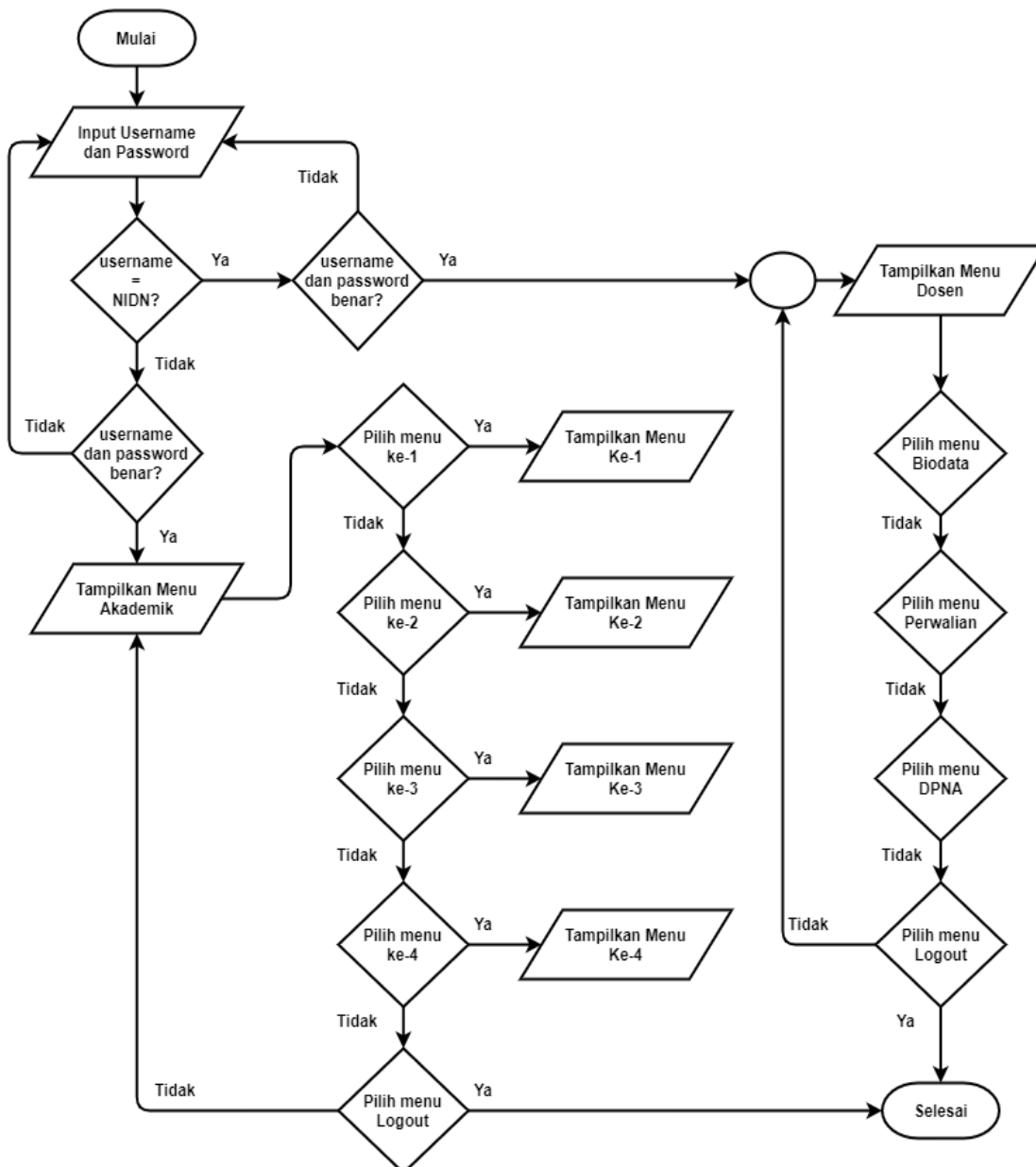
Ketemu \leftarrow false

Selama (tidak ketemu) dan ($i < N$) kerjakan baris 4 Jika ($Data[i] = key$) maka ketemu \leftarrow true

Jika tidak $i \leftarrow i + 1$

Jika (Ketemu) maka i adalah indeks dari data yang dicari

Flowchart sangat dibutuhkan untuk melihat setiap bagan aliran dalam suatu prosedur logika program[10]. Dimana *flowchart* digunakan untuk memberi keterangan jalur bagaimana logika terjadinya *input*, proses dan *output* dalam sebuah kegiatan. Berikut merupakan rancangan flowchart:



Gambar 4. Flowchart sistem aplikasi MyAKAD



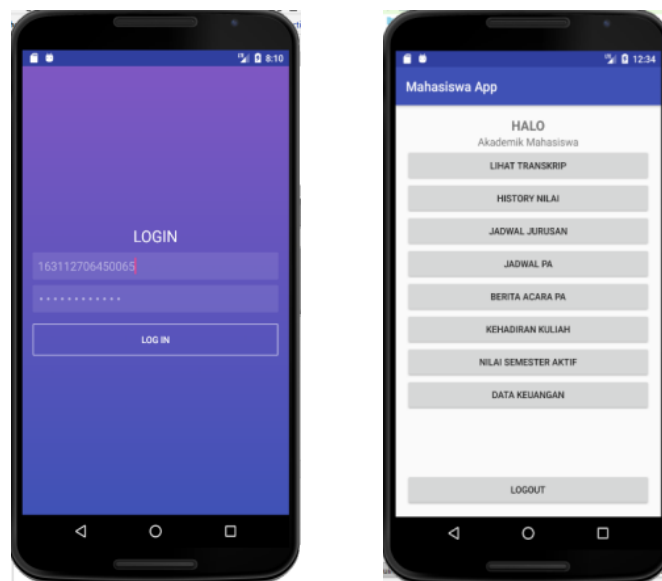
3.1 Rancangan Antarmuka

Tahap ini adalah proses pembuatan desain tampilan dan rancangan *input*, *process* dan *output* aplikasi dalam *platform website* dan pada *platform android* menggunakan bahasa pemrograman *PHP* dan *Java*.



Gambar 5. Tampilan Interface Website

Halaman ini terdapat *form login user* yang terdiri dari *form username* dan *password* untuk *login* ke MyAKAD Apps. Setelah *login*, *user* yang berhasil *login* akan berada di tampilan beberapa menu layanan administrasi akademik. (aplikasi masih dalam tahap pengembangan).



Gambar 6. Tampilan Interface Android

Di halaman *login* terdapat kolom *input NPM* dan *password* untuk akses ke layanan administrasi akademik. Jika *NPM* dan *password* melakukan *input* dengan benar (*true*), pengguna (*user*) dapat mengakses menu layanan akademik. Halaman menu utama untuk *user* terdiri dari: Lihat Transkrip, History Nilai, Jadwal Jurusan, Jadwal PA dan lain-lain.

3.2 Proses Kerja Algoritma Sequential Searching

Data yang terdapat pada *database* akan disimpan ke dalam variabel array. Di dalam variabel array data yang tersimpan merupakan NPM Mahasiswa. Penelitian ini hanya mengacu pada pencarian nomor NPM yang ada di dalam *database*. Berikut proses dan tahapan algoritma *sequential searching* :

```
a. <?php if(isset($_POST['login']))
{
$host = "localhost";
$username = "root";
$password = "";
$db = "db_myakad";
```



```
$db_link = mysqli_connect($host,$username,$password,$db);
```

(Script diatas berfungsi untuk melakukan koneksi ke database).

```
b. if (!$db_link){
```

```
echo "koneksi gagal";
```

```
}
```

(Script diatas berfungsi untuk mengecek apakah sudah terkoneksi dengan database atau belum).

```
c. $result = mysqli_query($db_link, "SELECT * FROM mahasiswa");
```

(Script diatas berfungsi untuk melakukan query pada tabel mahasiswa agar mendapatkan daftar mahasiswa).

```
d. $array = array();
```

```
$data = array();
```

(Script diatas berfungsi untuk menginisialisasikan variabel dengan tipe data array).

```
e. while ($row=mysqli_fetch_assoc($result))
```

```
{
```

```
$array[] = $row;
```

```
}
```

(Script diatas berfungsi untuk menempatkan atau menyimpan data yang diambil dari database ke dalam variabel array).

```
f. $xs = $_POST['npm'];
```

(Script diatas berfungsi untuk menginisialisasikan nilai yang akan dicari, dalam hal ini merupakan NPM mahasiswa).

```
g. $jumlah_mahasiswa = $array.size();
```

(Script diatas berfungsi untuk mendapatkan jumlah mahasiswa yang akan disimpan ke dalam variabel jumlah mahasiswa).

```
h. for ($i = 0; $i <= $jumlah_mahasiswa; $i++) { if ($array[$i] == $xs) {
```

```
$data['result'] = true;
```

```
} else {
```

```
$data['result'] = false;
```

```
}
```

```
}
```

(Script diatas berfungsi untuk melakukan pengulangan selama kurang dari sama dengan nilai yang tersimpan dalam variabel jumlah mahasiswa. Dilanjutkan dengan melakukan pengecekan apakah elemen array ke \$i merupakan sama dengan \$xs, Jika nilai tersebut sama. Maka akan memberi nilai true, false untuk sebaliknya).

3.3 White Box Testing

Pengujian ini sangat perlu dilakukan untuk memeriksa dan mengecek seluruh kode sumber pada perangkat lunak dan jalur logika secara valid. Lalu menentukan jumlah nilai untuk jalur independen pada perhitungan siklomatis kompleksitas.

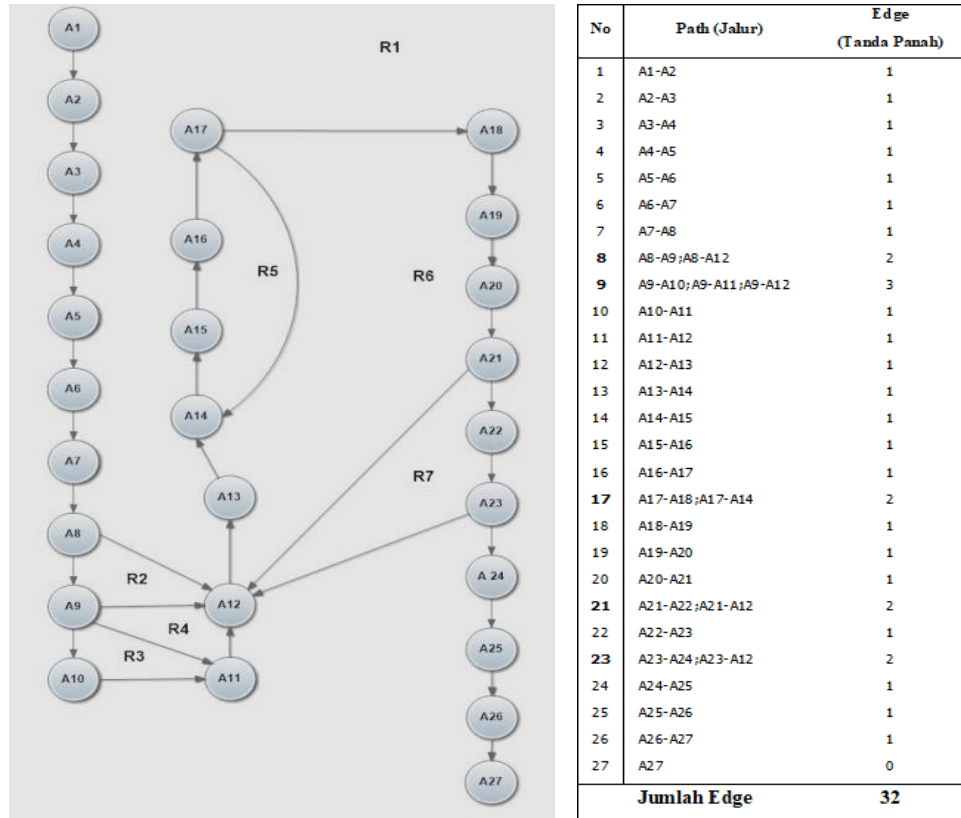
a. Index.php

A1	<?php	A15	{
A2	if(isset(\$_POST['login']))	A16	\$array[] = \$row;
A3	{	A17	}
A4	\$host = "localhost";	A18	\$xs = \$_POST['npm'];
A5	\$username = "root";	A19	\$jumlah_mahasiswa = (integer)count(\$data);
A6	\$password = "";	A20	for (\$i = 0; \$i <= \$jumlah_mahasiswa; \$i++) {
A7	\$db = "db_myakad";	A21	if (\$array['npm'] == \$xs) {
A8	\$db_link =	A22	\$data['result'] = true;
	mysqli_connect(\$host,\$username,\$password,\$db);	A23	}else{
A9	if (!\$db_link){		\$data['result'] = false;
		A24	}
A10	echo "koneksi gagal";	A25	echo json_encode(\$data);
A11	}	A26	}
A12	\$result = mysqli_query(\$db_link, "SELECT * FROM	A27	?>
	mahasiswa");		
A13	\$array = array();		
	\$data= array();		
A14	while (\$row=mysqli_fetch_assoc(\$result))		



b. *Basis Path Testing*

Digunakan untuk mendapatkan ukuran kompleksitas logika. Ukuran ini dijadikan sebagai panduan untuk menentukan jalur-jalur utama untuk dieksekusi. Path (jalur) adalah node (titik simpul) yang mungkin untuk bergerak dari satu titik ke lainnya sejalan dengan keberadaan arah anak panah (edge).



Gambar 7. Flowgraph dan Jumlah Edge

Dilihat hasil pengujian pada *basis path testing* pada gambar dibawah ini bahwa jalur yang di tentukan oleh node **A1** ke **A2** hanya bisa mengeksekusi **1 pernyataan**, sedangkan node **A8** jalurnya sebanyak **2 pernyataan** yaitu ke **A8 ke A9** dan **A8 ke A12** untuk dieksekusi, lalu **A9** sebanyak **3 pernyataan** yaitu jalur **A9 ke A10**, selanjutnya **A9 ke A11** dan **A9 ke A12** untuk di eksekusi dan seterusnya. Hasil dari penjumlahan *edge* (tanda panah) secara keseluruhan sangat dibutuhkan untuk perhitungan *cyclometri complexity* yang bertujuan menentukan jalur independen baru dalam pengujian *test case* yang menjamin semua *path* telah melakukan eksekusinya dengan benar.

c. *Cyclometri Complexity*

Perhitungan ini digunakan untuk menentukan jumlah independent path yang akan ditelusuri.

- Jumlah Region = R1,R2,R3,R4,R5,R6 dan R7 = 7
- Rumus Cyclometri Complexity = $V(G) = (E = Edge - N = Node) + 2$
 $= V(G) = E - N + 2$
 $= V(G) = 32 - 27 + 2 = 7$

Tabel 2. Perhitungan Predicate Node

No	Node (Titik Simpul)	Pencabangan	Jumlah Keputusan
1	A8	A12	1
2	A9	A11,A12	2
3	A17	A14	1
4	A21	A12	1
5	A23	A12	1
Total Keputusan (Predicate Node)			6

Predicate Note = Kondisi dimana node memiliki karakteristik dua atau lebih edge darinya.

Dimana P = Jumlah Predicate Node pada grafik alir (Total = 6).

- Rumus Cyclometri Complexity = $V(G) = P + 1$
 $= V(G) = (P = Predicate Node) + 1$



$$= V(G) = 6 + 1 = 7$$

Keterangan : “Total keputusan (Predicate Node) adalah hasil dari penjumlahan keputusan. 1 keputusan terdiri atas 2 percabangan dari 1 node (titik simpul). Total keputusan berjumlah 6, Perolehan jumlah keputusan tersebut sesuai dengan standar, maka algoritma yang digunakan cukup baik dan dapat menjamin bahwa semua jalur independen pada suatu modul sudah digunakan paling tidak satu kali. Keputusan *logis* dari sisi *true* ataupun *false* dan *looping* dapat tereksekusi secara keseluruhan. Dalam hal ini jumlah keputusan mewakili bahwa semua proses telah “**valid**” sesuai standar penelitian”.

Kesimpulan : Jika semua hasil nilainya sama pada saat *testing*, maka kompleksitas siklomatisnya “**valid**”.

d. Test Case

1. **Jalur 1 :** A1-A2-A3-A4-A5-A6-A7-A8-A12-A13-A14-A15-A16-A17-A18-A19-A20-A21-A22-A23-A24-A25-A26-A27.
2. **Jalur 2 :** A1-A2-A3-A4-A5-A6-A7-A8-A9-A12-A13-A14-A15-A16-A17-A18-A19-A20-A21-A22-A23-A24-A25-A26-A27.
3. **Jalur 3 :** A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A12-A13-A14-A15-A16-A17-A18-A19-A20-A21-A22-A23-A24-A25-A26-A27.
4. **Jalur 4 :** A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A11-A12-A13-A14-A15-A16-A17-A18-A19-A20-A21-A22-A23-A24-A25-A26-A27.
5. **Jalur 5 :** A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A11-A12-A13-A14-A15-A16-A17(A17-A14)-A18-A19-A20-A21-A22-A23-A24-A25-A26-A27.
6. **Jalur 6 :** A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A11-A12-A13-A14-A15-A16-A17(A17-A14)-A18-A19-A20-A21(A21-A12)-A22-A23-A24-A25-A26-A27.
7. **Jalur 7 :** A1-A2-A3-A4-A5-A6-A7-A8-A9-A10-A11-A12-A13-A14-A15-A16-A17(A17-A14)-A18-A19-A20-A21(A21-A12)-A22-A23(A23-A12)-A24-A25-A26-A27.

Keterangan : *Path* 1 sampai dengan 7 pada *test case* adalah hasil perhitungan kompleksitas siklomatis dalam merumuskan pencarian jalur independen. Hasil *test case* diatas menjamin semua jalur independen (*independent path*) telah dilakukan *testing* secara keseluruhan, dimana setiap *path* (jalur) memiliki eksekusi lebih dari 1 kali. Gambaran *test case* diatas diperlihatkan *path* pada *node* **A8** ke **A12** tidak hanya melakukan 1 kali eksekusi jalur, namun dapat melakukan lebih dari 1 kali, Sehingga diketahui ada 7 jalur (*path*) yang memiliki eksekusi lebih 1 kali pada beberapa *node*.

4. KESIMPULAN

Pembuatan aplikasi di *platform android* ini, meninjau kembali terhadap keterbatasan aplikasi administrasi di salah satu Universitas yang telah diteliti permasalahan sebelumnya, yang dimana aplikasi sebelumnya untuk akses ke menu layanan akademik harus memblokir pop-up terlebih dahulu, sedangkan aplikasi *MyAKAD Apps* berbasis *Android* ini dapat akses ke menu layanan akademik, tanpa harus memblokir pop-up. Hasil pengujian melalui white box testing menunjukkan total keputusan (predicate node) berjumlah 6 dari jumlah keputusan semua node (titik simpul) yang memiliki percabangan lebih dari satu path (jalur). Hasil nilai 7 pada perhitungan cyclomatic complexity telah sama dan sinkron dengan jumlah region. Dengan ini menyatakan bahwa semua proses pengujian algoritma telah valid sesuai hasil pengujian. Rancangan sistem berlaku untuk tingkat pengujian perangkat lunak seperti: Pengujian pada Unit, Pengujian pada Aplikasi, dan Pengujian Integrasi.

REFERENCES

- [1] L. S. Musmini and N. K. Kertiasih, “Sistem Administrasi Akademik Jurusan D3 Akuntansi Berbasis Teknologi Informasi,” *J. Pendidik. Teknol. dan Kejur.*, vol. 13, no. 2, pp. 225–232, 2016, doi: 10.23887/jptk.v13i2.8530.
- [2] W. F. Prasetyo, I. W. Sudarsana, and D. Lusiyanti, “Rancang Bangun Sistem Informasi Akademik Fmipa Universitas Tadulako Berbasis Android,” *J. Ilm. Mat. Dan Terap.*, vol. 15, no. 2, pp. 157–172, 2018, doi: 10.22487/2540766x.2018.v15.i2.11350.
- [3] H. Setiana and S. Hansun, “Gamified Android Based Academic Information System,” *Int. J. Eval. Res. Educ.*, vol. 6, no. 2, p. 164, 2017, doi: 10.11591/ijere.v6i2.7595.
- [4] M. Rizal and F. Latifah, “Perancangan Aplikasi Lokasi Wisata Kota Jakarta Menggunakan Algoritma Sequential Search Berbasis Android,” *None*, vol. 13, no. 2, pp. 227–232, 2017.
- [5] G. W. Sasmito, “Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal,” *J. Inform. Pengemb. IT*, vol. 2, no. 1, pp. 6–12, 2017.
- [6] D. Setiawan, T. Suratno, and Lutfi, “Analisis, Desain dan Rancang Bangun Sistem Informasi Akademik Mobile Berbasis Android,” *Elkha*, vol. 10, no. 2, p. 73, 2018, doi: 10.26418/elkha.v10i2.28272.
- [7] D. R. Sharma, H. B. Desai, K. C. Parmar, R. S. Sawant, and P. J. Gharat, “NotifyMe – An Android Application for Student Service Administration,” vol. 6, no. 3, pp. 297–300, 2017.
- [8] A. Sonita and M. Sari, “Implementasi Algoritma Sequential Searching Untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik,” *Pseudocode*, vol. 5, no. 1, pp. 1–9, 2018, doi: 10.33369/pseudocode.5.1.1-9.



- [9] A. Verma, A. Khatana, and S. Chaudhary, "A Comparative Study of Black Box Testing and White Box Testing," *Int. J. Comput. Sci. Eng.*, vol. 5, no. 12, pp. 301–304, 2017, doi: 10.26438/ijcse/v5i12.301304.
- [10] D. Yuzistin, D. F. Aji, and P. D. A. Pamungkas, "Sistem Informasi Administrasi Siswa Berbasis Website Pada SMA Islam Putradarma Bekasi," *Bina Insa. Ict J.*, vol. 3, no. 1, pp. 253–268, 2016.
- [11] I. Irawan, "PENGEMBANGAN SISTEM INFORMASI AKADEMIK UNIVERSITAS PAHLAWAN TUANKU TAMBUSAI RIAU," vol. 1, no. 2, pp. 55–66, 2018.
- [12] K. Anam and A. T. Muharram, "ANALISA DAN PERANCANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEB PADA MI AL-MURSYIDIYYAH AL- ' ASYIROTUSSYAFI ' IYYAH," vol. 11, no. 2, 2018.
- [13] M. Ibrohim and A. Hermawan, "Pengembangan Sistem Informasi Akademik Universitas Serang Raya (Infosera) Berbasis Mobile Android," pp. 41–46, 2016.