

Smart SIAKAD Berbasis Android Menggunakan *RecyclerView*

Smart SIAKAD Android based using RecyclerView

Azizah^{*1}, Iskandar Fitri², Fauziah³, Nur Hayati⁴

^{1,2}Universitas Nasional; Jl. Sawo Manila, Telp: (021) 7806700,

^{3,4}Jurusan Informatika, FTKI UNAS, Jakarta

e-mail: [*1azizahif99@gmail.com](mailto:azizahif99@gmail.com), [2iskandar.fitri@civitas.unas.ac.id](mailto:iskandar.fitri@civitas.unas.ac.id),

[3fauziah@civitas.unas.ac.id](mailto:fauziah@civitas.unas.ac.id), [4nurhayati@civitas.unas.ac.id](mailto:nurhayati@civitas.unas.ac.id)

Abstrak

Sistem informasi akademik (SIAKAD) berbasis android merupakan sistem informasi yang dapat mempermudah mahasiswa dan calon mahasiswa dalam mendapatkan informasi mengenai Universitas dan akademik melalui perangkat bergerak. Tahapan dalam penelitian menggunakan metode waterfall atau air terjun yang terdiri dari analisis, desain, pengodean, pengujian dan pemeliharaan. Aplikasi ini dibuat menggunakan metode RecyclerView dalam melakukan pengembangan aplikasi sistem informasi akademik dan melakukan pengujian diantaranya whitebox, blackbox, uji perangkat dan ui responden. Dari hasil pengujian whitebox mendapatkan total jumlah pada Cyclomatic Complex city (cc) sebesar 11, Region sebesar 15 dan independent Path sebesar 11. Pada uji coba blacblox didapatkan hasil bahwa aplikasi sudah dapat memberikan informasi sesuai kebutuhan yang ada. Pengujian menguji apakah fungsi dan tampilan dari aplikasi sudah layak untuk digunakan dan untuk uji coba perangkat didapatkan rata-rata waktu akses aplikasi yaitu 45 detik dan visual tampilan aplikasi 100%. Selain itu melakukan uji coba dengan responden sebanyak 23 responden dengan menguji aplikasi sudah berjalan dengan baik dengan hasil 52,2% setuju, 39,1% sangat setuju dan 8,7% tidak setuju. Pengembangan selanjutnya bisa diarahkan untuk penambahan fitur lain yang belum ada. Sebagai tambahan juga bisa dikembangkan untuk pengguna ponsel selain bersistem operasi Android seperti iOS.

Kata kunci — *Android, Sistem Informasi Akademik, aplikasi kampus, informasi kampus.*

Abstract

The Academic Information System (SIAKAD) android-based is an information system that can facilitate students and prospective students in obtaining information about the University and academics through mobile devices. The stages of study used waterfall method consisting of analysis, design, coding, testing and maintenance. This application is made using the RecyclerView method in conducting the application development of academic information systems and conducting tests such as Whitebox, blackbox, test devices and UI respondents. The results from the test of the whitebox, the total number of Cyclomatic Complex city (cc) amounted to 11, the Region of 15 and the independent Path of 11. The trial of the blacblox got the result that the app is already able to provide information to your existing needs. The tests of testing whether the function and appearance of the app is feasible to use and for a device trial it is obtained the average app access time of 45 seconds and the visual application display 100%. The others trial is conduct with respondents as many as 23 respondents by testing the application has been running well with the results 52.2% agreed, 39.1% very agreed and 8.7% disagree. Further of development can be directed to the addition of other features that don't exist yet. In addition, it can also be developed for mobile users in addition to Android operating systems such as iOS.

Keywords — *Android, academic information systems, campus applications, campus information.*

1. PENDAHULUAN

Sistem informasi merupakan suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisa, dan menyebarkan informasi untuk mencapai suatu tujuan [1]. Sistem informasi akademik yang digunakan universitas biasanya masih menggunakan sistem berbasis web yang mengharuskan pengunjung mengetik terlebih dahulu alamat website jika ingin mengunjungi sistem informasi tersebut. Disamping itu, penggunaan teknologi di indonesia khususnya *Smartphone* sangatlah tinggi. Berdasarkan laporan yang dikeluarkan oleh katadata.co.id, penggunaan *Smartphone* di Indonesia mencapai 92 juta unit [2]. Android adalah sistem operasi untuk *smartphone* yang banyak digunakan saat ini. Kelebihan android dengan sistem operasi *smartphone* lainnya adalah android bersifat *opensource code* sehingga memudahkan para pengembang untuk menciptakan dan memodifikasi aplikasi atau fitur-fitur yang belum ada sesuai dengan keinginan sendiri [3]. Pada penelitian sebelumnya mengenai aplikasi sistem informasi akademik pada tahun 2018 oleh Sri marinim telah membuat aplikasi *sistem* informasi akademik unisma berbasis android namun baru menerapkan mengenai profil, krs, dan khs [4]. Kemudian hasil penelitian yang dilakukan oleh Abdul Hariyanto, Endang Sugiharti, Riza Arifudin pada tahun 2019 aplikasi sistem informasi akademik yang menerapkan pembelajaran (*e-learning*) dan berita dengan berbasis android [5]. Berdasarkan dari penelitian sebelumnya, perancangan aplikasi sistem informasi akademik (SIAKAD) menggunakan berbasis Android dengan melakukan pengembangan yang mengimplementasikan *RecyclerView* dan sudah terintegrasi dengan *e-learning*, informasi profil mahasiswa, pemilihan krs, dan pendaftaran calon mahasiswa baru yang terhubung dengan internet. Pada aplikasi ini sudah mengalami perkembangan dari penelitian sebelumnya yaitu dengan menerapkan *recyclerview* dan semua sistem yang berbasis *website* sudah berbasis Android.

2. METODE PENELITIAN

Dalam metode penelitian ini terdiri dari beberapa tahap yaitu, studi literatur, perumusan masalah, pengembangan sistem dan penarikan simpulan. Studi literatur adalah serangkaian kegiatan yang berkaitan dengan metode pengumpulan data pustaka, membaca dan mencatat, serta mengelolah bahan penelitian. Menurut Danial dan Warsiah Studi Literatur adalah merupakan penelitian yang dilakukan oleh peneliti dengan mengumpulkan sejumlah buku-buku, majalah yang berkaitan dengan masalah dan tujuan penelitian. Teknik ini dilakukan dengan tujuan untuk mengungkapkan berbagai teori-teori yang relevan dengan permasalahan yang sedang dihadapi atau diteliti sebagai bahan rujukan dalam pembahasan hasil penelitian [6]. Tahap perumusan masalah memperjelas permasalahan sehingga dapat mempermudah dalam menyelesaiakannya. Dari hasil studi literatur, penulis menemukan permasalahan yang dapat dirumuskan yaitu, Bagaimana membangun aplikasi sistem informasi akademik (*Smart SIAKAD*) berbasis Android. Selanjutnya pengembangan sistem dalam pengembangan sistem penulis memakai metode air terjun atau *waterfall* dengan alur terurut dimulai dari analisis, desain, pengodean, pengujian dan pemeliharaan [7].

Membangun aplikasi Android menjadi hal yang menarik dan menantang juga. Jadi sebelum kita siap untuk mencapai target maka mendesain aplikasi yang paling penting untuk memeriksa apakah itu mencapai kebutuhan yang diinginkan [8]. Selanjutnya pengembangan sistem dengan melakukan analisa kebutuhan sistem, desain, pembuatan program dan pengujian aplikasi. *RecyclerView* sebuah komponen tampilan (widget) yang lebih canggih dari pada pendahulunya *listview*. Ia bersifat lebih fleksibel. *RecyclerView* mempunyai kemampuan untuk

menampilkan data secara efisien dalam jumlah yang besar. Terlebih jika memiliki koleksi data dengan elemen yang mampu berubah-ubah sewaktu dijalankan (runtime). [9]. Gambar 1 merupakan komponen *RecyclerView*.



Gambar 1. Komponen *RecyclerView*

2.1 Analisa Sistem

Menurut Kristanto (2003) analisis sistem merupakan suatu proses mengumpulkan dan menginterpretasikan kenyataan, mendiagnosa persoalan dan menggunakan keduanya untuk memperbaiki sistem. Menurut Yogyianto (1995) analis sistem (analisis informasi) merupakan orang yang menganalisis sistem (mempelajari masalah yang timbul dan menentukan kebutuhan pemakai sistem) untuk mengidentifikasi pemecahan permasalahan tersebut [10]. Aplikasi ini memiliki kategori pengguna yaitu mahasiswa, dosen dan umum. Setiap kategori memiliki kebutuhan sistem masing-masing diantaranya :

- Kebutuhan Umum yaitu dapat melihat informasi, fasilitas kampus, dan daftar kuliah.
- Kebutuhan mahasiswa yaitu dapat login ke akademik *online* serta webkuliah untuk dapat mengetahui khs,krs dan lainnya.
- Kebutuhan dosen yaitu untuk mengakses akademik *online* dan webkuliah untuk penginputan nilai, mengajar dan lainnya.

Untuk itu dalam membuat aplikasi ini kebutuhan yang digunakan adalah.

Tabel 1. Perangkat Lunak

No	Perangkat Lunak
1	OS Windows 10 64bit
2	Android Studio 3.4
3	Google Chrome
4	Figma

Dari penguraian Tabel 1 bertujuan dalam pembuatan aplikasi tidak terjadi proses yang lama dalam penginputan data dan bisa menghasilkan kinerja yang maksimal sesuai dengan kebutuhan *user* (pemakai). Sedangkan pada Tabel 1 telah disebutkan beberapa perangkat lunak (*software*) yang mendukung dalam pembuatan aplikasi tersebut.

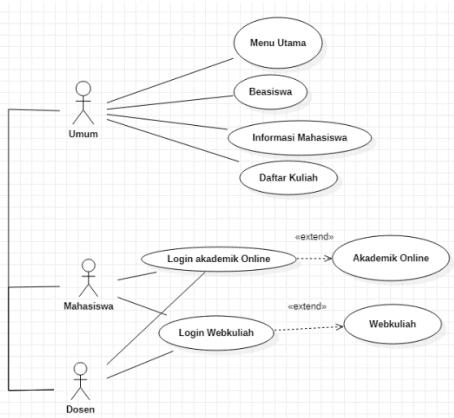
Tabel 2. Perangkat Keras

No	Perangkat Keras
	1 Buah Laptop (ASUS VivoBook A442U) dengan spesifikasi sebagai berikut :
1.	a. CPU Intel Core i5 8 th Gen
	b. RAM 8GB DDR4
	c. HDD 1TB
	d. NVIDIA 930MX

Pada Tabel 2 telah disebutkan beberapa perangkat keras yang dibutuhkan dalam pembuatan aplikasi.

2.2 Desain Sistem

Use case diagram untuk fitur yang ada di aplikasi *Smart SIAKAD*



Gambar 2. Use case diagram Aplikasi *Smart SIAKAD*

Gambar 2 terdapat 3 aktor yaitu umum, mahasiswa dan dosen. Dimana pengguna umum dapat mengakses beberapa fitur seperti fasilitas kampus, informasi mahasiswa baru, daftar kuliah dan yang lainnya. Untuk fitur akademik *online* dan webkuliah hanya bisa diakses oleh mahasiswa dan dosen. Penjelasan mengenai use case diagram dijelaskan di tabel 3.

Tabel 3. Penjelasan use case.

Nama Use Case	Deskripsi	Aktor
Melihat Menu utama	Aktor dapat melihat menu utama	Mahasiswa, dosen dan umum
Melihat Beasiswa	Aktor dapat melihat list beasiswa	Mahasiswa, dosen dan umum
Melihat Informasi mahasiswa baru	Aktor dapat melihat informasi mahasiswa baru	Mahasiswa, dosen dan umum
Daftar kuliah	Aktor dapat daftar kuliah	Mahasiswa, dosen dan umum
Login akademik <i>online</i>	Aktor dapat login kedalam akun akademik dengan akun yang ditentukan	Mahasiswa dan dosen
Login webkuliah	Aktor dapat login kedalam akun webkuliah	Mahasiswa dan dosen

2.3 Coding (Pengkodean)

Penerapan kode program pada aplikasi *Smart SIAKAD* ini menggunakan pemrograman Java Android yang menerapkan POJO pada salah satu menu yang diimplementasikan menggunakan *RecyclerView*.

2.4 Testing (Pengujian)

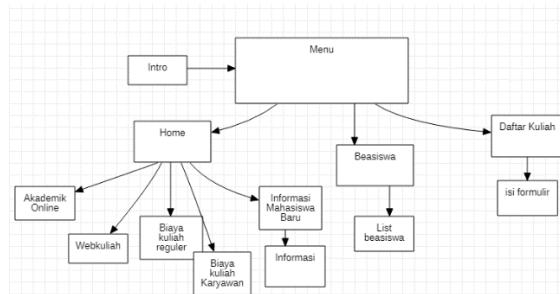
Penulis menggunakan uji coba menggunakan metode blackbox dan hasil uji coba di evaluasi dengan memanfaatkan data kuisioner dari responden.

2.5 Pemeliharaan

Pada tahap pemeliharaan dilakukan perbaikan aplikasi berdasarkan evaluasi dari pengguna atau pembimbing, kemudian memperbaruiinya.

3. HASIL DAN PEMBAHASAN

Dalam Gambar 3 menunjukan tentang *user flow* dari aplikasi *Smart SIAKAD*. Dari gambar tersebut kita mengetahui bahwa *user* pertama kali melihat pengenalan aplikasi kemudian baru bisa masuk ke menu utama. Penulis membahas mengenai implementasi *RecyclerView* pada salah satu menu di aplikasi *Smart SIAKAD* yaitu di menu Beasiswa. *RecyclerView* yang digunakan yaitu dengan tampilan kartu (*CardView*) dan *LayoutManager*: Komponen antarmuka yang bertugas untuk menampilkan data set yang dimiliki di dalamnya mengatur posisi tampilan data secara list (vertikal)



Gambar 3. *User Flow*

```
implementation 'androidx.recyclerview:recyclerview:1.0.0'
implementation 'androidx.cardview:cardview:1.0.0'
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp"
        android:layout_marginBottom="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:background="#f0f0f0"
        android:clipToPadding="false"
        android:padding="10dp"
        android:scrollbars="vertical"
        android:stackFromBottom="true"
        android:stackFromEnd="true"
        android:transcriptMode="alwaysScroll"
        android:verticalScrollbarPosition="left"
        android:verticalScrollBarWidth="10dp"
        tools:listitem="@layout/item_recycler_view" />

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

Gambar 4. Menambahkan dependencies. Gambar 5. Menambahkan *RecyclerView*.

Pada gambar 4 Menambahkan dependencies komponen *RecyclerView* pada file build.gradle sedangkan pada gambar 5 Menambahkan *RecyclerView* di berkas tampilan xml dari fragment beasiswa.

```
public class Beasiswa implements Parcelable {
    private String name;
    private String desk;
    private String lok;
    private int photo;

    public int getPhoto() { return photo; }
    public void setPhoto(int photo) { this.photo = photo; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getDesk() { return desk; }
    public void setDesk(String desk) { this.desk = desk; }
    public String getLok() { return lok; }
    public void setLok(String lok) { this.lok = lok; }

    @Override
    public int describeContents() { return 0; }

    @Override
    public void writeToParcel(Parcel dest, int flags) {
        dest.writeString(this.name);
        dest.writeString(this.desk);
        dest.writeString(this.lok);
        dest.writeInt(this.photo);
    }

    public Beasiswa() {
    }
}
```

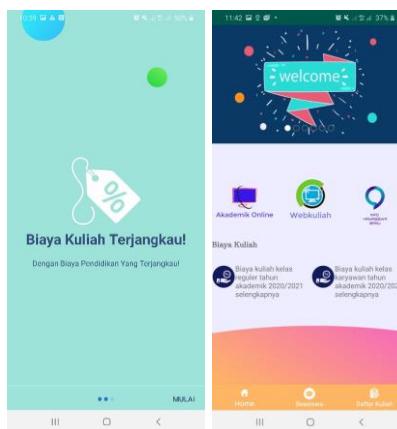
Gambar 6. Model Kelas Beasiswa.

Pada gambar 6 menerapkan model kelas POJO dan Parcelable pada data beasiswa.

Gambar 7. Berkas layout xml baris item. Gambar 8. Membuat Kelas Adapter.

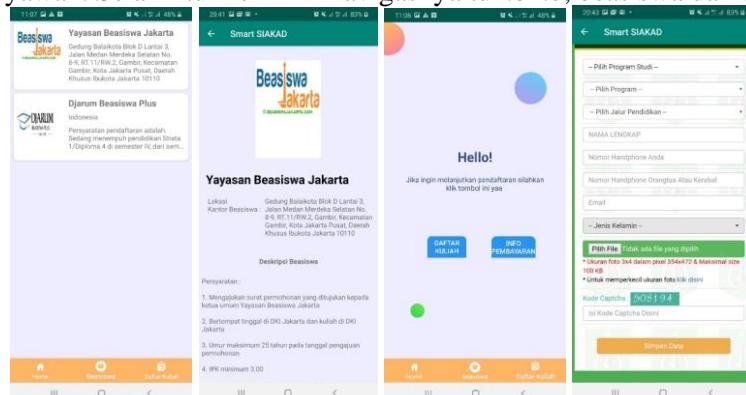
Gambar 7 membuat berkas layout xml untuk baris item di *RecyclerView*. Pada gambar 8 membuat sebuah kelas adapter yang inherit ke *RecyclerView*. Adapter dan *ViewHolder* untuk menampilkan tiap elemen data. Berikutnya mendefinisikan obyek *RecyclerView* berikut dengan bentuk yang diinginkan (bisa dalam bentuk *list*, *grid*, atau *staggered*) dan selanjutnya pasang obyek adapter (*binding*) agar bisa menampilkan koleksi data ke dalam *RecyclerView*.

3.1. Simulasi atau Testing Aplikasi *Smart SIAKAD*



Gambar 9. Tampilan Pengenalan dan Menu Home

Gambar 9 merupakan pengenalan aplikasi dan menu *home* dari aplikasi *Smart SIAKAD*. Pengenalan aplikasi ini mengenalkan secara singkat kampus. Menu *Home* atau menu utama menampilkan mengenai fitur yang dapat digunakan di aplikasi *Smart SIAKAD* yang berisi menu akademik *online*, web kuliah, informasi mahasiswa baru, biaya kuliah regular dan biaya kuliah karyawan. Selain itu memiliki navigasi yaitu *home*, beasiswa dan daftar kuliah.



Gambar 10. Tampilan Menu Beasiswa dan Daftar Kuliah

Pada gambar 10 merupakan tampilan dari menu beasiswa dan daftar kuliah. Pada menu beasiswa merupakan beasiswa yang bisa diajukan oleh mahasiswa. Pada menu ini menjelaskan persyaratan untuk bisa mendapatkan beasiswa tersebut. Tampilan menu daftar kuliah yang terdiri dari daftar kuliah dan info pembayaran. Pada menu daftar kuliah calon mahasiswa baru bisa mendaftarkan diri dengan mengisi formulir yang tersedia. Jika sudah mengisi calon mahasiswa bisa melihat menu info pembayaran untuk langkah selanjutnya.

Pada hasil penelitian yang telah ditunjukkan sebelumnya mengalami pengembangan yaitu dengan adanya informasi mengenai suatu beasiswa yang menerapkan *recyclerview* serta pada aplikasi ini dalam mengakses profil mahasiswa, pembelajaran kuliah serta pendaftaran kuliah sudah berbasis Android.

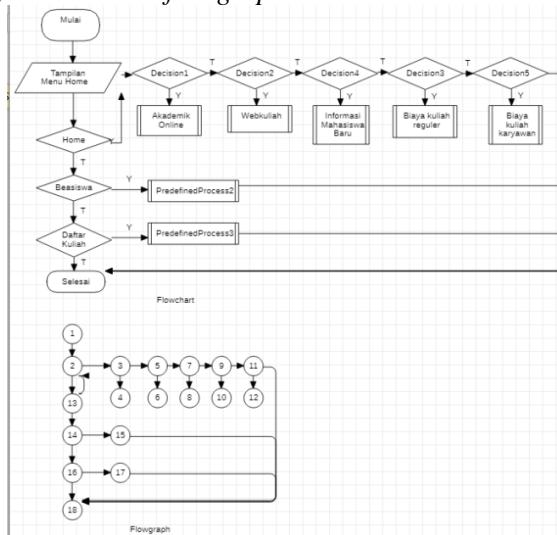
3.2. Pengujian

Setelah aplikasi selesai dibuat diperlukan suatu pengujian untuk mengetahui aplikasi yang dibuat sudah sesuai dengan tujuan yang diharapkan. Pada pengujian aplikasi SIAKAD menggunakan teknik *whitebox testing* dan *blackbox testing*.

Pengujian dilakukan dengan menguji setiap proses dan kemungkinan kesalahan yang terjadi untuk setiap proses. Adapun pengujian sistem yang digunakan adalah *White Box testing* melibatkan pengetahuan teknis terperinci dari sistem. Untuk menguji *software*, tester membuat pengujian yang paling struktural dengan melihat kode dan struktur data itu sendiri [11].

a. *Flowchart dan Flowgraph* Tampilan Menu Home

Gambar 11 merupakan *flowchart* dan *flowgraph* Menu home



Gambar 11. *Flowchart* dan *flowgraph* Menu Home.

Dari gambar 12 merupakan *flowchart* dan *flowgraph* Menu home dapat dilakukan proses sebagai berikut .

Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$E(\text{edge}) = 21 \quad N(\text{node}) = 18$$

$$V(G) = E - N + 2$$

$$= 21 - 18 + 2$$

$$= 5$$

Jadi, jumlah *Path* dari *flowgraph* di atas sebanyak 5 *Path*

Menghitung berdasarkan *Predicate Node* (P)

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus $V(G) = P + 1$ dimana $P = 8$

$$V(G) = P + 1$$

$$= 8 + 1$$

$$= 9$$

Jumlah *Region* (R) yang terdapat dalam *flowgraph* di atas adalah 9.

Path-Path yang terdapat dalam *Flowgraph* di atas

Path 1 = 1 – 2 – 3 – 5 – 7 – 9 – 11 – 18

Path 2 = 1 – 2 – 13 – 2 – 3 – 5 – 7 – 9 – 11 – 18

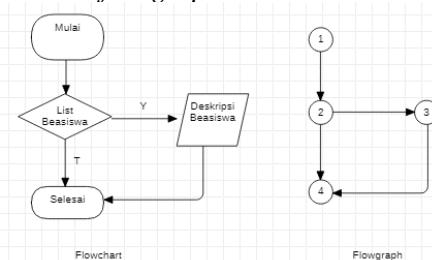
Path 3 = 1 – 2 – 13 – 14 – 15 – 18

Path 4 = 1 – 2 – 13 – 14 – 16 – 17 – 18

Path 5 = 1 – 2 – 13 – 14 – 16 – 18

b. *Flowchart* dan *Flowgraph* Tampilan Menu Beasiswa

Gambar 13 merupakan *flowchart* dan *flowgraph* Menu beasiswa



Gambar 12. *Flowchart* dan *flowgraph* Menu Beasiswa.

Dari gambar 12 merupakan *flowchart* dan *flowgraph* Menu beasiswa dapat dilakukan proses sebagai berikut .

Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$E(\text{edge}) = 4 \quad N(\text{node}) = 4$$

$$V(G) = E - N + 2$$

$$= 4 - 4 + 2$$

$$= 2$$

Jadi, jumlah *Path* dari *flowgraph* di atas sebanyak 2 *Path*

Menghitung berdasarkan *Predicate Node* (P)

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus $V(G) = P + 1$ dimana $P = 1$

$$V(G) = P + 1$$

$$= 1 + 1$$

$$= 2$$

Jumlah *Region* (R) yang terdapat dalam *flowgraph* di atas adalah 2.

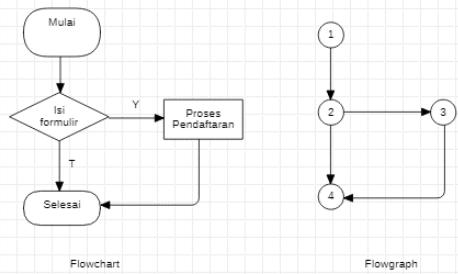
Path-Path yang terdapat dalam *Flowgraph* di atas

$$\text{Path 1} = 1 - 2 - 3 - 4$$

$$\text{Path 2} = 1 - 2 - 4$$

c. *Flowchart* dan *Flowgraph* Tampilan Menu Daftar Kuliah

Gambar 14 merupakan *flowchart* dan *flowgraph* Menu daftar kuliah



Gambar 13. *Flowchart* dan *flowgraph* Menu Daftar Kuliah.

Dari gambar 13 merupakan *flowchart* dan *flowgraph* Menu daftar kuliah dapat dilakukan proses sebagai berikut .

Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$E(\text{edge}) = 4 \quad N(\text{node}) = 4$$

$$V(G) = E - N + 2$$

$$= 4 - 4 + 2$$

$$= 2$$

Jadi, jumlah *Path* dari *flowgraph* di atas sebanyak 2 *Path*.

Menghitung berdasarkan *Predicate Node* (P)

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus $V(G) = P + 1$ dimana $P = 1$

$$V(G) = P + 1$$

$$= 1 + 1$$

$$= 2$$

Jumlah *Region* (R) yang terdapat dalam *flowgraph* di atas adalah 2.

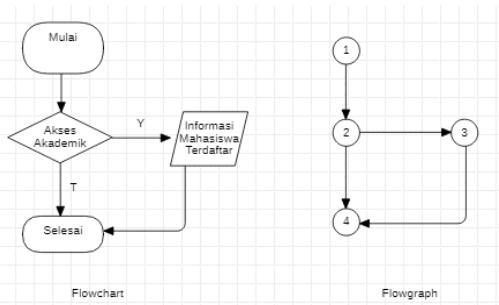
Path-Path yang terdapat dalam *Flowgraph* di atas

$$\text{Path 1} = 1 - 2 - 3 - 4$$

$$\text{Path 2} = 1 - 2 - 4$$

d. *Flowchart* dan *Flowgraph* Tampilan Menu Akademik *Online*

Gambar 15 merupakan *flowchart* dan *flowgraph* Menu akademik *online*



Gambar 14. *Flowchart* dan *flowgraph* Menu Akademik *Online*.

Dari gambar 14 merupakan *flowchart* dan *flowgraph* Menu akademik *online* dapat dilakukan proses sebagai berikut .

Menghitung *Cyclomatic Complexity* dari *Edge* dan *Node*

$$E(\text{edge}) = 4 \quad N(\text{node}) = 4$$

$$V(G) = E - N + 2$$

$$= 4 - 4 + 2$$

$$= 2$$

Jadi, jumlah *Path* dari *flowgraph* di atas sebanyak 2 *Path*.

Menghitung berdasarkan *Predicate Node* (P)

P adalah jumlah titik yang menyatakan logika dalam diagram alir dengan rumus $V(G) = P + 1$ dimana $P = 1$

$$V(G) = P + 1$$

$$= 1 + 1$$

$$= 2$$

Jumlah *Region* (R) yang terdapat dalam *flowgraph* di atas adalah 2.

Path-Path yang terdapat dalam *Flowgraph* di atas

Path 1 = 1 – 2 – 3 – 4

Path 2 = 1 – 2 – 4

Tabel 4. Rekapitulasi Hasil Pengujian *whitebox*.

Flowgraph	Cyclomatic Complex city (cc)	Region(R)	Independen Path
Menu Home	5	9	5
Menu Beasiswa	2	2	2
Menu Akademik	2	2	2
Menu Daftar Kuliah	2	2	2
Jumlah	11	15	11

Pada tabel 4 ditemukan nilai *Cyclomatic Complexcity* (cc) sebesar 11 yang artinya sistem memiliki resiko menengah.

Pengujian *blackbox* ini dilakukan untuk memperlihatkan fungsi aplikasi yang dibuat tentang cara operasi dan kegunaannya, apakah keluaran data sesuai dengan yang diharapkan. Hasil pengujian aplikasi secara keseluruhan dengan menggunakan metode *blackbox* dapat dilihat pada Tabel 5.

Tabel 5. Pengujian aplikasi dengan metode *blackbox testing*.

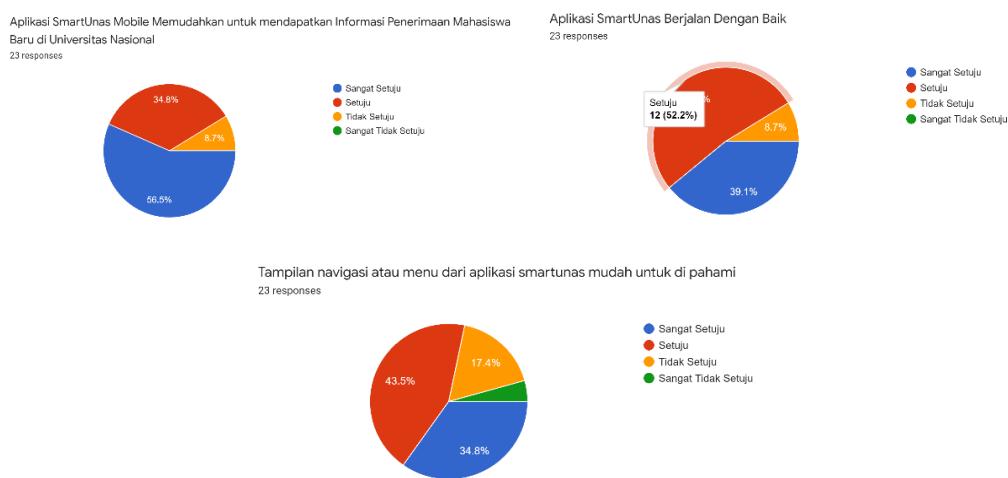
Pengujian	Bentuk Pengujian	Hasil yang diharapkan	Hasil pengujian
Menu Home	Masuk ke halaman utama	Terdapat menu akademik, webkuliah, informasi mahasiswa baru, biaya kuliah regular dan karyawan. Navigasi home, beasiswa dan daftar kuliah	Berhasil
Menu Akademik <i>online</i>	Menekan tombol akademik <i>online</i>	Tampilan akademik <i>online</i>	Berhasil
Menu Beasiswa	Menekan navigasi beasiswa	Tampilan list beasiswa	Berhasil
Menu Daftar Kuliah	Menekan navigasi daftar kuliah	Tampilan daftar kuliah	Berhasil

Pengujian perangkat juga dibutuhkan untuk mengetahui apakah aplikasi sudah layak dan kompatibel di perangkat yang diujikan.

Tabel 6. Pengujian Perangkat

Perangkat	Sistem Operasi	Waktu Akses Aplikasi	Visual
Samsung Galaxy A50	Android 10.0 Q	40 detik	100%
Samsung Galaxy A10	Android 9.0 Pie	31 detik	100%
Samsung J2 Prime	Android 6.0.1 Marshmellow	30 detik	100%
Samsung S4	Android 5.0.1 Lollipop	69 detik	100%
Samsung Galaxy A6	Android 9.0 Pie	54 detik	100%
Samsung J7 2016	Android 7.0 Nougat	49.44 detik	100%

Pada table 6 menunjukan bahwa tampilan visual aplikasi sudah berjalan 100% dan waktu akses aplikasi dengan rata-rata 45 detik untuk dapat mengakses aplikasi SIAKAD dari pengenalan hingga menutup aplikasi. Setelah melakukan uji coba dengan menggunakan metode *blackbox*, *Whitebox* dan uji perangkat selanjutnya dilakukan pengujian kepada pengguna sebagai responden. Pengujian ini bertujuan untuk melihat bahwa sistem dapat memenuhi kriteria kebutuhan atau dapat diterima oleh pengguna aplikasi..



Gambar 15. Hasil pengujian kepada responden.

Pada gambar 15 merupakan hasil pengujian pertama dengan aplikasi memudahkan untuk mendapatkan informasi penerimaan mahasiswa baru di kampus yang diikuti oleh responden sebanyak 23 responden dengan hasil 34,8% setuju, 56,5% sangat setuju dan 8,7% tidak setuju. Hasil pengujian kedua dengan aplikasi berjalan dengan baik yang diikuti oleh responden sebanyak 23 responden dengan hasil 52,2% setuju, 39,1% sangat setuju dan 8,7% tidak setuju. Hasil pengujian ketiga dengan tampilan navigasi atau menu dari aplikasi mudah untuk di pahami yang diikuti oleh responden sebanyak 23 responden dengan hasil 43,5% setuju, 34,8% sangat setuju, 17,4% tidak setuju dan 4,3 sangat tidak setuju.

4. KESIMPULAN

Aplikasi SIAKAD merupakan aplikasi sistem informasi akademik kampus yang memiliki fitur seperti akademik *online* yaitu merupakan sistem akademik mahasiswa yang bisa mengakses khs, krs, transkrip dan yang lannya. Selain itu aplikasi ini juga dapat mengakses webkuliah yang merupakan sistem e-learning. Di aplikasi ini juga terdapat menu daftar kuliah yang bisa digunakan untuk calon mahasiswa baru yang ingin mendaftar sebagai mahasiswa. Aplikasi ini juga bisa diakses secara *offline* maupun *online*. Namun untuk sistem pembelajaran secara daring hanya dapat diakses menggunakan internet. Dari hasil pengujian *whitebox* mendapatkan total jumlah pada *Cyclomatic Complex city (cc)* sebesar 12, *Region* sebesar 16 dan *independent Path* sebesar 12. Pada uji coba *blackbox* didapatkan hasil bahwa aplikasi sudah dapat memberikan informasi sesuai kebutuhan yang ada. Pengujian menguji apakah fungsi dan

tampilan dari aplikasi sudah layak untuk digunakan dan untuk uji coba perangkat didapatkan rata-rata waktu akses aplikasi yaitu 45 detik dan visual tampilan aplikasi 100%. Selain itu melakukan uji coba dengan responden sebanyak 23 responden dengan menguji aplikasi sudah berjalan dengan baik dengan hasil 52,2% setuju, 39,1% sangat setuju dan 8,7% tidak setuju.

5. SARAN

Diharapkan pada penelitian selanjutnya dapat mengembangkan sistem dengan mengintegrasikan sistem media sosial kampus agar antar mahasiswa dapat melakukan *sharing*. Selain itu, diharapkan sistem dapat dikembangkan dari sisi platform dari berbasis android ditambahkan menjadi iOS agar semakin memudahkan pengguna untuk mengakses sistem.

DAFTAR PUSTAKA

- [1]. Ananda, Nadia Arifi. 2016. Sistem Informasi Akademik SMA Roudlotul Ummah Berbasis Android. Jurnal Transit. Volume-, No.-, Agustus.
- [2]. Admin, “Pengguna Smartphone di Indonesia 2016-2019 | Databoks,” 2019. [Daring]. Tersedia pada: <https://databoks.katadata.co.id/datapublish/2016/08/08/pengguna-Smartphone-di-indonesia-2016-2019>. [Diakses: 16 April 2020].
- [3]. Sarbini, Riska Nurtantyo, dkk. 2016. Rancang Bangun Sistem Informasi Akademik Berbasis Android. Jurnal Dedikasi, ISSN 1693-3214.
- [4]. Marini, Sri. 2016. Rancang Bangun Sistem Informasi Akademik Unisma Berbasis Android dan Web. Prosiding Seminar Nasional Energi & Teknologi(SINERGI).
- [5]. Haryanto, Abdul, dkk. 2019. Aplikasi *Mobile* Sistem Informasi Akademik Labschool Universitas Negeri Semarang Berbasis Android. UNNES Journal of Mathematics 8(1).
- [6]. N. Fatin, “Pengertian Studi Literatur.” [Daring]. Tersedia pada: <http://seputarpengertian.blogspot.com/2017/09/pengertian-studi-literatur.html>. [Diakses: 16 April 2020].
- [7]. Rosa, A.S. 2011. Rekayasa Perangkat Lunak. Bandung: Modula.
- [8]. 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.
- [9]. Dicoding, “Belajar Membuat Aplikasi Android untuk Pemula | RecyclerView Teori,” 2017. Tersedia pada: <https://www.dicoding.com/academies/51/tutorials/1224>. [Diakses: 16 April 2020].
- [10]. Jony. 2015. Aplikasi Informasi Akademik Berbasis Android. Jurnal SISFOKOM, Volume 04, Nomor 02, September.
- [11]. Hidayat, M. Irwan. 2016. Aplikasi Monitoring Aktivitas Santri Berbasis (Studi Kasus : Pesantren Modern Ulul Al-Bab Makassar). Fakultas Sains dan Teknologi UIN Alauddin Makassar.