**SKRIPSI**

**APLIKASI AGENDA BERBASIS ANDROID DENGAN FITUR PUSH NOTIFICATION DAN REMINDER**

**(Studi Kasus : UKM INFORMATIKA & KOMPUTER)**

****

**RESTA BAYU SETIAWAN**

**Nomor Mahasiswa : 145410161**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER AKAKOM**

**YOGYAKARTA**

**2018**

**HALAMAN PERSETUJUAN**

JUDUL : APLIKASI AGENDA BERBASIS ANDROID DENGAN

FITUR PUSH NOTIFICATION DAN REMINDER

(Studi Kasus : UKM INFORMATIKA & KOMPUTER).

NAMA : RESTA BAYU SETIAWAN

NIM : 145410161

JURUSAN : TEKNIK INFORMATIKA

SEMESTER : GANJIL 2018/2019

Telah memenuhi syarat dan disetujui untuk diuji di hadapan dosen penguji pendadaran tugas akhir.

Yogyakarta, Januari 2019

Mengetahui

Dosen Pembimbing

Indra Yatini B, S.Kom., M.Kom.

**HALAMAN PENGESAHAN**

**SKRIPSI**

**APLIKASI AGENDA BERBASIS ANDROID DENGAN FITUR PUSH NOTIFICATION DAN REMINDER**

**(Studi Kasus : UKM INFORMATIKA & KOMPUTER)**

Telah dipertahankan di depan dewan penguji tugas akhir dan dinyatakan diterima untuk memenuhi sebagai syarat guna memperoleh gelar Sarjana Komputer Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM

Yogyakarta,\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mengesahkan

Dewan Penguji Tanda Tangan

1. Indra Yatini B, S.Kom, M.Kom.
2. L.N. Harnaningrum, S.Si, M.T.
3. Drs. Tri Prabawa, M.Kom.

Mengetahui,

Ketua Program Studi Teknik Informatika

Dini Fakta Sari S.T, M.T.

**HALAMAN PERSEMBAHAN**

Skripsi ini saya persembahkan untuk :

~

Mamakku Samilah yang selalu memberikan cinta dan kasih sayang pada diriku melebihi apapun yang beliau tau.

~ ~

Bapakku Ngatiman yang selalu berkerja keras dengan segala kemampuanya untuk menghidupi keluarga dan membiayai diriku sampai lulus menjadi sarjana.

~ ~ ~

Kedua Adikku Sila dan Vellia yang selalu ada dihatiku.

~ ~ ~ ~

Kerabat dan saudara yang tak henti-hentinya menanyakan kepada diriku, “kapan saya lulus” ?.

~ ~ ~ ~ ~

Dan terakhir untuk diri kecilku ini yang sudah berusaha hingga menyelesaikan tugas mulia ini dengan amanah dan jujur.

**MOTTO**

Mentari terbit saat pagi dan terbenam diwaktu sore. Hujan turun setelah mendung dan berhenti sebelum pelangi. Ketahuilah bahwa semua hal mempunyai waktu dan ritme nya tersendiri, begitu pula dengan dirimu. Maka jangan pernah membandingkan dirimu dengan siapapun atau apapun.

(Penulis)

Jangan menepi, jangan berhenti, tetap terkini, tetap terkini !

(Penulis)

**INTISARI**

Unit Kegiatan Mahasiswa Informatika & Komputer (UKM IK) merupakan salah satu organisasi mahasiswa yang ada di STMIK AKAKOM Yogyakarta yang memiliki berbagai agenda kegiatan internal baik yang bersifat akademik maupun non akademik. Penyampaian agenda kegiatan yang akan diselenggarakan tersebut saat ini masih menggunakan media Whatsapp yang dianggap kurang efektif karena harus mengirim satu-persatu kepada seluruh anggota yang ada, dan kadangkala ada beberapa anggota yang tidak mengikuti acara tersebut karena lupa akan informasi yang telah didapatkan tempo hari.

Dengan adanya permasalahan tersebut maka dilakukan pengembangan aplikasi yang dapat mengirimkan agenda kegiatan secara serempak kepada anggota UKM IK sekaligus dapat memberikan pengingat otomatis kepada anggota apabila sudah memasuki waktu dimana acara yang ada akan segera dimulai.

Aplikasi ini berbasis android yang dibangun menggunakan bahasa pemrograman Java, XML sebagai tampilan antarmukanya dan diterapkan pula teknologi Firebase Authentication, Firebase Cloud Messaging, Firebase Realtime Database untuk membuat aplikasi ini dapat diandalkan

**KATA PENGANTAR**

Puji syukur atas kehadirat Tuhan Yang Maha Esa, berkat kehendaknya penulis dapat menyelesaikan skripsi yang berjudul “APLIKASI AGENDA BERBASIS ANDROID DENGAN FITUR PUSH NOTIFICATION DAN REMINDER”. Sehingga penulis dapat memenuhi salah satu syarat untuk lulus dari program studi S1 Teknik Informatika STMIK AKAKOM YOGYAKARTA.

Penulisan skripsi ini dapat terselesaikan atas bantuan dari berbagai pihak. Maka dengan ini penulis ingin menyapaikan ucapan terimaksih kepada :

1. Bapak Ir. Totok Suprawoto, M.M, M.T. selaku ketua umum STMIK AKAKOM YOGYAKARTA
2. Ibu Dini Fakta Sari S.T, M.T. selaku ketua jurusan Teknik Informatika STMIK AKAKOM YOGYAKARTA
3. Ibu Indra Yatini B, S.Kom, M.Kom. selaku dosen pembimbing skripsi yang senantiasa membimbing dan memberikan arahan kepada penulis dalam mengegerjakan skripsi ini hingga selesai.
4. Ibu L.N. Harnaningrum, S.Si, M.T. selaku dosen narasumber skripsi yang senantiasa mengkritisi dan memberi masukkan kepada penulis.
5. Bapak Drs. Tri Prabawa, M.Kom. selaku dosen narasumber yang senantiasa mengkritisi dan memberi masukkan kepada penulis.
6. Sahabatku Asep Ahmad Sofyan yang senantiasa membantu mengatasi masalah teknis dari salah fitur yang ada pada aplikasi penelitian ini.
7. Teman-teman yang selalu mensupport dan memberikan semangat pada penulis untuk terus berusaha.

Semoga skirpsi ini dapat memberikan manfaat khususnya kepada Akademisi STMIK AKAKOM dan UKM Informatika dan Komputer selaku organisasi yang menjadi objek dalam penelitian ini.

Yogyakarta, 31 Desember 2018

Resta Bayu Setiawan

**DAFTAR ISI**

HALAMAN PERSETUJUAN i

HALAMAN PENGESAHAN ii

HALAMAN PERSEMBAHAN iii

MOTTO iv

INTISARI v

KATA PENGANTAR vi

DAFTAR ISI viii

DAFTAR GAMBAR x

DAFTAR TABEL xi

BAB I. PENDAHULUAN 1

1.1. Latar Belakang Masalah 1

1.2. Rumusan Masalah 2

1.3. Ruang Lingkup 2

1.4. Tujuan Penelitian 3

1.5. Manfaat Penelitian 4

BAB II. TINJAUAN PUSTAKA DAN DASAR TEORI 5

2.1. Tinjauan Pustaka 5

2.2. Dasar Teori 8

2.2.1. Unit Kegiatan Mahasiswa Informatika dan Komputer

(UKM IK) 8

2.2.2. Android 9

2.2.3. Firebase 9

2.2.4. Firebase Authentication 10

2.2.5. Firebase Cloud Messaging 10

2.2.6. Firebase Realtime Database 11

BAB III. METODE PENELITIAN 12

3.1. Bahan / Data 12

3.2. Peralatan 12

3.2.1. Kebutuhan Perangkat Lunak 12

3.2.2. Kebutuhan Perangkat Keras 13

3.3. Perancangan Sistem 13

3.3.1. Arsitektur Sistem 13

3.3.2. Use Case Diagram 14

3.3.3. Sequence Diagram 15

3.3.4. Activity Diagram 17

3.3.5. Class Diagram 20

3.3.6. Perancangan Antarmuka 21

BAB IV. IMPLEMENTASI DAN PEMBAHASAN 26

4.1. Implementasi Sistem 26

4.2. Ujicoba dan Pembahasan 37

4.2.1. Menambahkan Acara Baru dari Aplikasi Admin 38

4.2.2. Halaman Registrasi Aplikasi Pengguna 39

4.2.3. Halaman Login Aplikasi Pengguna 40

4.2.4. Halaman Utama Aplikasi Pengguna 42

4.2.5. Halaman Detail Acara Aplikasi Pengguna 43

4.2.6. Notifikasi Acara Baru 44

4.2.7. Notifikasi Pengingat Acara 45

BAB V. KESIMPULAN DAN SARAN 47

5.1. Kesimpulan 47

5.2. Saran 47

DAFTAR PUSTAKA 49

**DAFTAR GAMBAR**

Gambar 3.1. Arsitektur Sistem 13

Gambar 3.2. Use Case Diagram Admin 14

Gambar 3.3. Use Case Diagram User 14

Gambar 3.4. Sequence Diagram Tambah Acara Baru 15

Gambar 3.5. Sequence Diagram Edit Data Acara 16

Gambar 3.6. Sequence Diagram Login User 16

Gambar 3.7. Sequence Diagram Lihat Acara 17

Gambar 3.8. Activity Diagram Tambah Acara Baru 18

Gambar 3.9. Activity Diagram Edit Data Acara 18

Gambar 3.10. Activity Diagram Login User 19

Gambar 3.11. Activity Diagram Lihat Acara 20

Gambar 3.12. Class Diagram 20

Gambar 3.13. Tampilan Login Aplikasi 21

Gambar 3.14. Tampilan Registrasi Pengguna 22

Gambar 3.15. Tampilan Utama Daftar Acara 23

Gambar 3.16. Tampilan Detail Acara 23

Gambar 3.17. Tampilan Admin Daftar Acara 24

Gambar 3.18. Tampilan Kelola Data Acara 25

Gambar 4.1. Menambahkan Data Acara 27

Gambar 4.2. Melihat Daftar Acara 28

Gambar 4.3. Mengirim Push Notification 29

Gambar 4.4. Mengedit Acara 30

Gambar 4.5. Menghapus Data Acara 31

Gambar 4.6. Registrasi 32

Gambar 4.7. Login 32

Gambar 4.8. Mengambil Data Dari Firebase Realtime Database 33

Gambar 4.9. Menampilkan Daftar Acara Pada Halaman Utama 34

Gambar 4.10. Halaman Detail Acara 35

Gambar 4.11. Pengampu Push Notification 36

Gambar 4.12. Menambahkan Pengingat Acara 37

Gambar 4.13. Menambahkan Acara 38

Gambar 4.14. Daftar Acara Yang Ada 39

Gambar 4.15. Halaman Registrasi Aplikasi Pengguna 40

Gambar 4.16. Halaman Login Aplikasi Pengguna 41

Gambar 4.17. Halaman Slide Show Aplikasi Pengguna 42

Gambar 4.18. Halaman Utama Aplikasi Pengguna 43

Gambar 4.19. Halaman Detail Acara Aplikasi Pengguna 44

Gambar 4.20. Notifikasi Acara Baru 45

Gambar 4.21. Notifikasi Pengingat Acara 46

**DAFTAR TABEL**

Tabel 2.1. Tinjauan Pustaka 7

# **BAB I**

**PENDAHULUAN**

1. **Latar Belakang Masalah**

Unit Kegiatan Mahasiswa Informatika & Komputer (UKM IK) merupakan salah satu organisasi mahasiswa yang ada di STMIK AKAKOM Yogyakarta. Dalam menjalanakan perannya sebagai organisasi mahasiswa UKM IK memiliki struktur organisasi yag terdiri atas Pengurus Harian (PH), Dewan Penasihat Organisasi (DPO), dan Anggota. UKM IK memiliki berbagai kegiatan internal baik yang bersifat akademik maupun non akademik seperti *Study Club*, Rapat, Kunjungan Industri dan Kumpul Bareng.

Dalam menginformasikan agenda kegiatan yang akan diselenggarakan tersebut, bagian PH Departement Hummas biasanya memberitahukan kepada seluruh anggota UKM IK melalui media *whatsapp* *Mesenger* secara personal. Proses penyebaran informasi yang ada tersebut dianggap kurang efektif, karena Hummas harus mengirimkan pesan satu-persatu kepada seluruh anggota & terkadang ada anggota yang tidak bisa mengikuti kegiatan karena lupa akan informasi yang sudah didapatkan tempo hari.

Dari permasalahan yang ada tersebut penulis berencana membuat penelitian dengan membangun aplikasi yang dapat meninformasikan agenda acara yang akan diselenggarakan oleh UKM IK. Aplikasi ini berbasis Android yang nantinya dapat diinstall pada masing-masing smartphone anggota UKM IK. Pengguna aplikasi ini dapat melihat agenda kegiatan berupa informasi waktu, tempat, dan deskripsi acara yang akan diselenggarakan oleh UKM IK. Adanya fitur *push notification* juga akan membuat pengguna aplikasi ini menerima notifikasi setiap ada agenda kegiatan baru yang telah ditambahkan pada aplikasi. Selain itu akan ditambahkan *reminder* pada aplikasi ini yang akan mengingatkan pengguna apabila memasuki hari acara tersebut akan diselenggarakan.

1. **Rumusan Masalah**

Berdasarkan latar belakang masalah yang ada diatas maka rumusan masalah dalam penelitian ini yaitu : Bagaimana langkah membangun aplikasi agenda kegiatan UKM IK berbasis android dengan fitur *push notification* dan *reminder.*

1. **Ruang Lingkup**

Agar pembahasan pada penelitian ini nantinya tidak melebar maka diberikan ruang lingkup seperti berikut :

1. Aplikasi ini dibangun menggunakan *Android Studio* IDE dan *Firebase.*
2. Bahasa pemrograman yang digunakan adalah Java.
3. Pengguna aplikasi ini adalah seluruh anggota UKM IK.
4. Informasi kegiatan yang dimuat dalam aplikasi ini hanya kegiatan internal yang akan diselenggarakan oleh UKM IK seperti Rapat Anggota, *Study Club*, Kunjungan Industri dan Kumpul Bareng.
5. Update informasi kegiatan hanya bisa dilakukan oleh admin.
6. Anggota UKM IK akan menerima notifikasi apabila ada info kegiatan baru pada aplikasi.
7. Anggota UKM IK dapat melihat daftar kegiatan yang akan diselenggarakan oleh UKM IK.
8. Anggota UKM IK akan menerima *reminder* apabila memasuki hari acara yang ada akan diselenggarakan.
9. Untuk dapat mengakses aplikasi ini dibutuhkan koneksi internet.
10. Untuk dapat menjalankan semua fitur pada aplikasi ini digunakan *smartphone* android dengan sistem operasi minimal android 5.0 *Lollipop*
11. **Tujuan Penelitian**

Tujuan dari penelitian ini adalah untuk menghasilkan rancangan aplikasi android yang bisa menampilkan informasi kegiatan UKM IK dan memberikan *reminder* ke pengguna apabila memasuki hari pada saat acara akan dimulai.

1. **Manfaat Penelitian**

Manfaat dari penelitian ini adalah untuk membuat proses penyebaran informasi agenda kegiatan UKM IK kepada seluruh anggota menjadi lebih efisien.

# **BAB II**

**TINJAUAN PUSTAKA DAN DASAR TEORI**

1. **Tinjauan Pustaka**

Dalam tinjauan pustaka ini diawali dengan menelaah penelitian terdahulu yang memiliki keterkaitan serta relevansi dengan penelitian yang akan dilakukan. Degan demikian maka dapat diambil rujukan pendukung untuk membuat penelitian ini menjadi lebih memadai.

Pada tahun 2017 Aghnia Fi’la Urfan melakukan penelitian dengan topik Aplikasi Kalender Event Seni Kontemporer, dimana objek dari penelitian tersebut adalah kegiatan Seni Kontemporer yang ada di wilayah Daerah Istimewa Yogyakarta. Dari penelitian tersebut dihasilkan aplikasi android Jogja Festivals. Aplikasi tersebut dibangun menggunakan Android Studio IDE, dimana bahasa pemrograman yang digunakan adalah java, *database* nya menggunakan *MySQL*, serta *Google Map API* sebagai *library* untuk fitur penunjuk arah. Fitur dari aplikasi ini pengguna dapat melihat daftar acara yang ada, menambahkan pengingat acara ke *Google Calendar*, serta melakukan tracking penunjuk arah tempat berlangsungnya acara dengan *Google Maps*.

Kemudian Mursito melakukan penelitian dengan topik Aplikasi Android Logbook Tugas Akhir Semester menggunakan *Firebase Cloud Messaging*. Objek dari penelitian tersebut adalah Bimbingan Tugas Akhir Skripsi Fakultas MIPA Universitas Negeri Yogyakarta. Dari penelitian

yang dilakukan oleh mursito dihasilkan aplikasi android Logbook TAS. Aplikasi tersebut dibangun Menggunakan Android Studio dengan bahasa pemrograman java, *NosSQL database*, dan *Firebase Cloud Messaging* untuk menghandel fitur notifikasi.

Reyhan Pradipta melakukan penelitian dengan topik Aplikasi *Mobile Notification* Perkuliahan. Dimana objek dari penelitian tersebut adalah perkuliahan di STMIK AKAKOM YOGYAKARTA. Dari penelitian tersebut dihasilkan aplikasi android Infoku. Aplikasi tersebut dibangun menggunakan Andorid Studio dengan bahasa pemrograman java, *MySQL database*, dan *Firebase Cloud Messaging* untuk menghandel fitur notifikasi. Fitur dari aplikasi tersebut adalah pengguna dapat melihat daftar matakuliah yang diambil serta menerima notifikasi apabila ada pengumuman mengenai matakuliah yang bersangkutan.

Reza Abdillah melakukan penelitian dengan topik Implementasi *Push Notification* pada Aplikasi Lowongan Kerja. Objek dari penelitian tersebut adalah Informasi Lowongan Pekerjaan di Daerah Istimewa Yogyakarta. Dari penelitian tersebut dihasilkan aplikasi android Loker Jogja. Aplikasi tersebut dibangun Menggunakan Android Studio dengan *library Parse, Parse NosSQL database,* dan *Google Cloud Messaging API*.

Silva Casiavera juga melakukan penelitian dengan topik Aplikasi Pengingat Jadwal Imunisasi. Objek dari penelitian tersebut adalah Kegiatan Imunisasi Balita di Kabupaten Bantul. Aplikasi tersebut dibangun menggunakan Android Studio dengan bahasa pemrograman java dan *MySQL* sebagai *databasenya*. Fitur dari aplikasi ini adalah orangtua sebagai pengguna dapat melihat jadwal imunisasi pada anak, dan menerima notifikasi pada saat memasuki hari jadwal imunisasi.

Berdasarkan tinjauan pustaka yang ada diatas, perbedaan aplikasi yang akan dibuat oleh penulis adalah terletak pada objek penelitian & teknologi yang digunakan untuk membuat aplikasi ini. Dimana objek yang digunakan untuk penelitian ini adalah organisasi mahasiswa UKM Informatika & Komputer STMIK AKAKOM YOGYAKARTA. Sedangkan teknologi yang digunakan oleh penulis antara lain adalah : *mobile android* dan *firebase.*

Tabel 2.1. Tinjauan Pustaka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nama Peneliti** | **Topik** | **Teknologi** | **Objek** | **Hasil Keluaran** |
| Aghnia Fi’la Urfan (2017) | Aplikasi Kalender Event Seni Kontemporer | Android, Integrasi Google Map & Google Calendar | Kegiatan Seni Kontemporer wilayah DIY | Aplikasi Android Jogja Festivals |
| Mursito (2017) | Aplikasi Loogbok Tugas Akhir Semester menggunakan Firebase Cloud Messaging | Android, Firebase Cloud Messaging | Bimbingan Tugas Akhir Skripsi Fakultas MIPA UNY | Aplikasi Android Logbook TAS |
| Reyhan Pradipta Sumardi (2017) | Aplikasi Mobile Notification Perkuliahan | Android, Firebase Cloud Messaging | Perkuliahan STMIK AKAKOM | Aplikasi Android Infoku |
| Reza Abdillah  (2017) | Implementasi Push Notification Aplikasi LOKER | Android, Parse, Google Cloud Messaging | Informasi Lowongan Pekerjaan di DIY | Aplikasi Android Loker Jogja |
| Silva Casiavera (2018) | Aplikasi Pengingat Jadwal Imunisasi | Android, MySQL | Imunisasi Balita di Kabupaten Bantul | Aplikasi Android Pengingat Imunisasi |
| Penelitian yang diusulkan (2018) | Aplikasi Agenda UKM IK | Android, Firebase Auth, Firebase Cloud Messaging, Firebase Realtime Database. | UKM Informatika & Komputer (UKM IK) | Aplikasi Android Agenda UKM IK |

1. **Dasar Teori**
   * 1. **Unit Kegiatan Informatika & Komputer (UKM IK)**

UKM IK merupakan salah satu Unit Kegiatan Mahasiswa di STMIK AKAKOM Yogyakarta. UKM IK didirikan pada tanggal 12 April 1995. Adapun alasan didirikannya UKM Informatika dan Komputer adalah untuk menampung aspirasi mahasiswa STMIK AKAKOM Yogyakarta dalam bidang penalaran ilmiah, serta sebagai sarana untuk menyalurkan minat dan bakat mahasiswa dalam bidang penalaran ilmiah.

Sejak awal berdiri hingga saat ini, UKM Informatika dan Komputer telah melalui banyak hal, mengalami pasang surut kehidupan organisasi. Namun itu tidak merubah orientasi UKM IK untuk tetap mengembangkan prestasi dalam bidang penalaran ilmiah dengan mengadakan kegiatan-kegiatan yang digunakan untuk menunjang kemampuan mahasiswa khususnya anggota UKM Informatika dan Komputer.

* + 1. **Android**

Android merupakan sistem operasi berbasis *linux* yang dikembangkan oleh *Google Inc*. untuk perangkat bergerak seperti telepon pintar dan komputer tablet. Antarmuka pengguna Android umumnya berupa manipulasi langsung dengan menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta manipulasi langsung dengan menggunakan papan ketik *virtual* untuk menulis teks.

Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Aplikasi yang selesai dibuat oleh pengembang ini nantinya dapat di *publish* ke *Google Play Store* selaku *marketplace* aplikasi resmi dari Android. (Menurut statista.com) Saat ini terdapat lebih dari 3 juta aplikasi android yang dapat diunduh di *Google Play Store*.

* + 1. **Firebase**

*Firebase* pertama kali didirikan pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk yang pertama kali dikembangkan adalah *Realtime Database*, di mana developer dapat menyimpan dan melakukan sinkronasi data ke banyak user. Kemudian pada Oktober 2014 Firebase telah diakusisi oleh Google. Berbagai fitur terus dikembangkan hingga saat ini berbagai produk layanan Firebase seperti *Firebase Realtime Database, Firebase Authentication, Firebase Cloud Messaging, Firebase Crash Report, Firebase Hosting* dan masih banyak lagi.

* + 1. **Firebase Authentication**

*Firebase Authentication* adalah layanan yang disediakan oleh *Firebase* untuk membangun sistem autentikasi yang aman, sekaligus meningkatkan pengalaman *login* dan pengalaman aktivasi bagi pengguna akhir. Fungsi ini menyediakan solusi identitas *end-to-end*, mendukung login dengan menggunakan akun email atau autentikasi nomor telepon.

*Firebase Authentication* menyediakan User Interface dengan solusi autentikasi yang dapat disesuaikan untuk menangani aliran *User Interface* pada *form login* pengguna akhir.

* + 1. **Firebase Cloud Messaging**

*Firebase Cloud Messaging* (FCM) adalah layanan dari *Firebase* yang menawarkan solusi *server push* untuk aplikasi Android. Server push maksudnya adalah fitur digunakan apabila aplikasi *server* kita ingin mentrigger aplikasi mobile. Dengan FCM memungkinkan pengembang untuk memberikan *push notification* dan membuat komunikasi dua arah antar device. Teknologi yang digunakan terbagi menjadi dua :

1. XMPP (Extensible Messaging and Presence Protocol)

Untuk XMPP pengembang harus membangun *server* XMPP terlebih dahulu.

1. HTTP (Hypertext Transfer Protocol)

Untuk HTTP pengembang bisa menggunakan console yang disediakan oleh Firebase.

* + 1. **Firebase Realtime Database**

*Firebase Realtime Database* adalah layanan basis data *NoSQL* yang disediakan oleh *Firebase* yang dapat digunakan untuk menyimpan dan menyinkronkan data secara real-time. *NoSQL* merupakan basis data yang tidak menggunakan sistem relasi layaknya basis data pada umumnya seperti *MySQL*. Metode penyimpanan data di dalam *NoSQL* menggunakan objek yang menggunakan format *JSON* *(JavaScript Object Notation).*

Dengan *Firebase Realtime Database* memungkinkan untuk menggunakan basis data yang ketika di *share* kepada semua user, dan ketika terjadi perubahan data pada basis data tersebut maka user akan segera mendapatkan update data secara *real time*.

# **BAB III**

**METODE PENELITIAN**

1. **Bahan / Data**
   * 1. **Kebutuhan Input**
2. Admin : admin mempunyai *useremail* dan *password* untuk melakukan login ke Firebase Console guna melihat traffik dan data pengguna maupun acara yang ada pada aplikasi.
3. Anggota : anggota mempunyai *useremail* dan *password* untuk *login* ke aplikasi.
   * 1. **Kebutuhan Output**

Kebutuhan output dari aplikasi adalah menampilkan daftar informasi kegiatan UKM IK yang dapat dilihat oleh pengguna aplikasi.

1. **Peralatan**
   * 1. **Kebutuhan Perangkat Lunak**

1. Sistem Operasi Linux Elementary OS Loki (64 bit)

2. Java JDK versi 8

3. Android Software Development Kit (SDK)

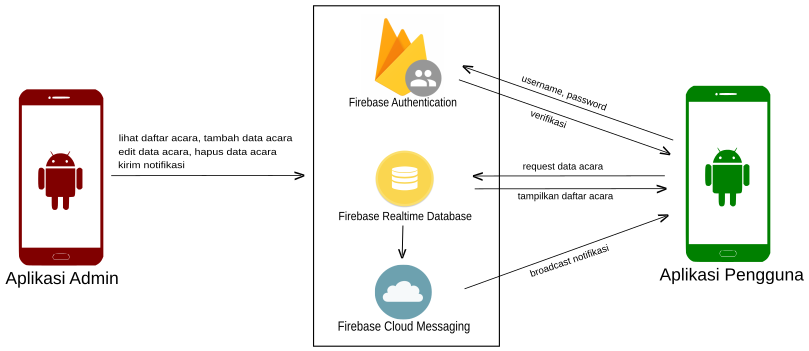
4. Android Studio Development Tools

5. Firebase Console

6. StarUML diagram editor

* + 1. **Kebutuhan Perangkat Keras**

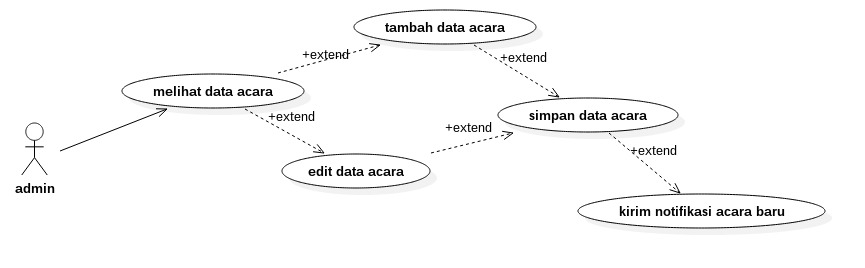
1. Laptop Dengan Spesifikasi
   1. Processor Intel Core i3 2.4 Ghz
   2. RAM 4 GB
   3. VGA Intel HD Graphics
   4. Hardisk 500 GB
2. *Smartphone* Android dengan spesifikasi
3. Layar 5 inchi
4. Processor Octa Core 1.4 Ghz
5. RAM 2 GB
6. Memori internal 32 GB
7. **Perancangan Sistem**
8. **Arsitektur Sistem**



Gambar 3.1. Arsitektur Sistem

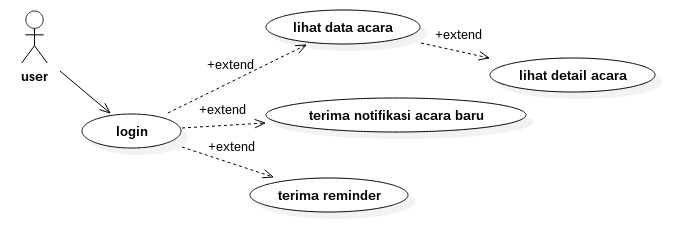
1. **Use Case Diagram**

Pada *Use Case Diagram* dibawah ini admin melakukan login terlebih dahulu, kemudian admin dapat melihat data acara yang ada. Setelah itu admin dapat menambah acara baru maupun mengedit data acara yang sudah ada kemudian menyimpannya. Setelah itu admin dapat melakukan *broadcast* notifikasi ke seluruh pengguna aplikasi.



Gambar 3.2. Use Case Diagram Admin

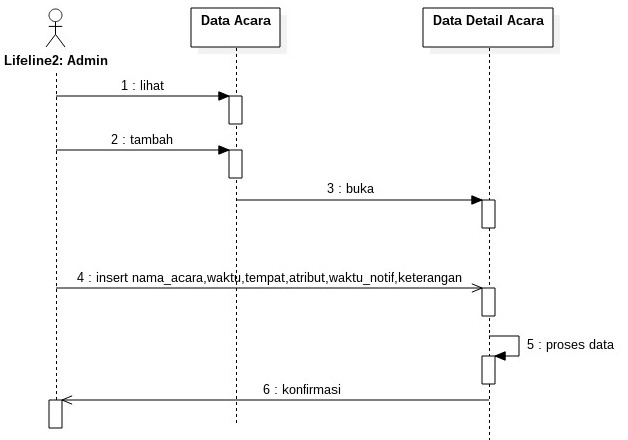
User melakukan login terlebih dahulu, setelah itu user dapat melihat daftar acara yang ada lalu melihat detail masing-masing acara tersebut. Kemudian user juga akan menerima notifikasi apabila ada acara baru. User juga akan menerima *reminder* apabila memasuki hari dimana acara tersebut akan dilaksanakan.



Gambar 3.3. Use Case Diagram User

1. **Sequence Diagram**
2. Sequence Diagram Tambah Acara Baru

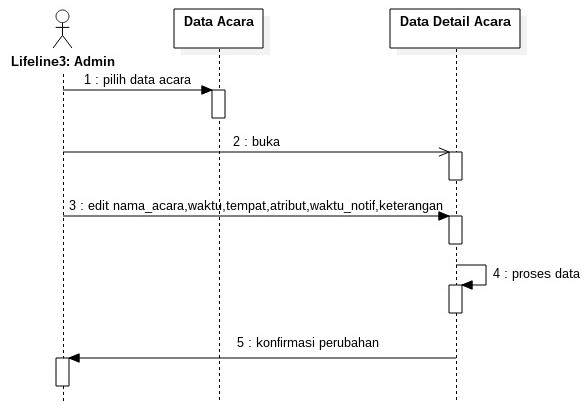
Pada bagian ini admin dapat melihat data acara yang ada. Kemudian admin dapat menambahkan acara baru dengan memasukkan nama\_acara, waktu, tempat, atribut, waktu\_notif, dan keterangan.



Gambar 3.4. Sequence Diagram Tambah Acara Baru

1. Sequence Diagram Edit Data Acara

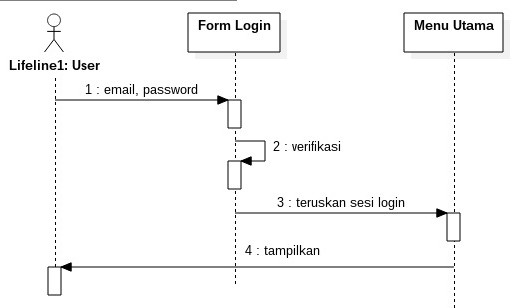
Pada bagian ini admin dapat memilih data acara yang sudah ada. Setelah itu admin dapat membuka detail data acara tersebut dan melakukan perubahan pada data yang ada.



Gambar 3.5. Sequence Diagram Edit Data Acara

1. Sequence Diagram Login User

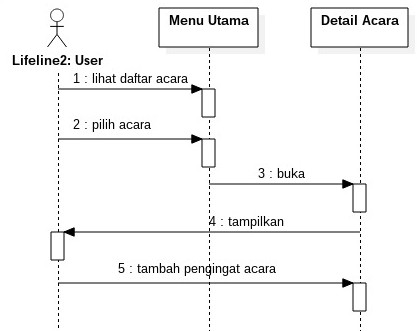
Pada bagian ini user dapat memasukkan *email* dan *password* di *form login* untuk melakukan login. Kemudian akan dilakukan verifikasi *login* oleh *form login*. Setelah verifikasi berhasil maka sesi *login* berhasil dan akan diteruskan ke menu utama.



Gambar 3.6. Sequence Diagram Login User

1. Sequence Diagram Lihat Acara

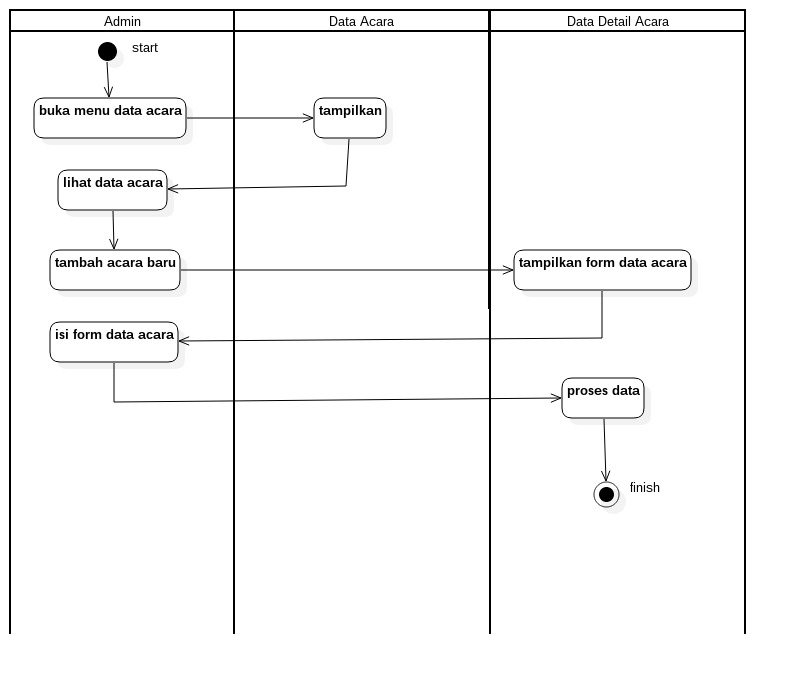
Pada bagian ini *user* dapat melihat daftar acara yang ada, kemudian *user* dapat memilih dan membuka detail acara tersebut, dan melihat informasi detail acara yang ada.



Gambar 3.7. Sequence Diagram Lihat Acara

1. **Activity Diagram**
2. Activity Diagram Tambah Acara Baru

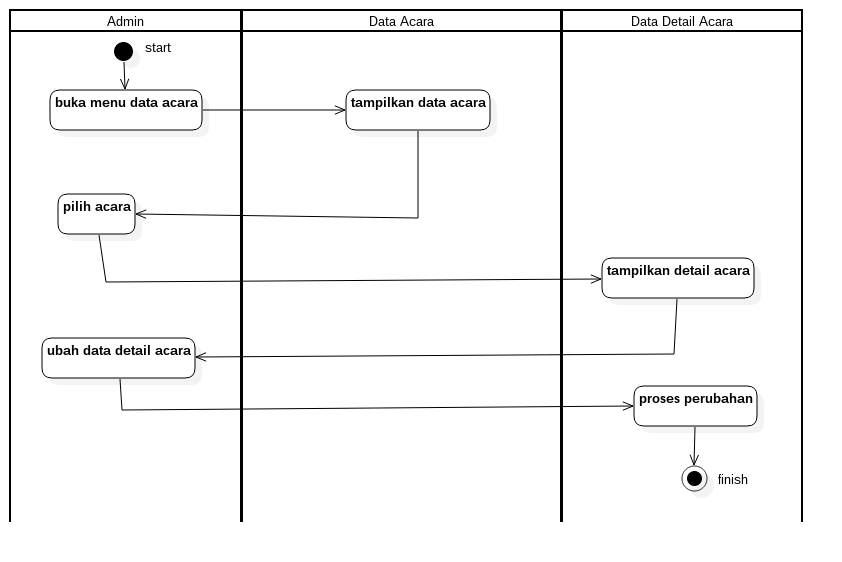
Pada bagian ini Admin admin bisa membuka menu data acara, kemudian admin dapat melihat data acara yang sudah ada. Admin juga dapat menambahkan acara baru dengan mengisikan rincian acara yang disediakan oleh aplikasi



Gambar 3.8. Activity Diagram Tambah Acara Baru

1. Activity Diagram Edit Data Acara

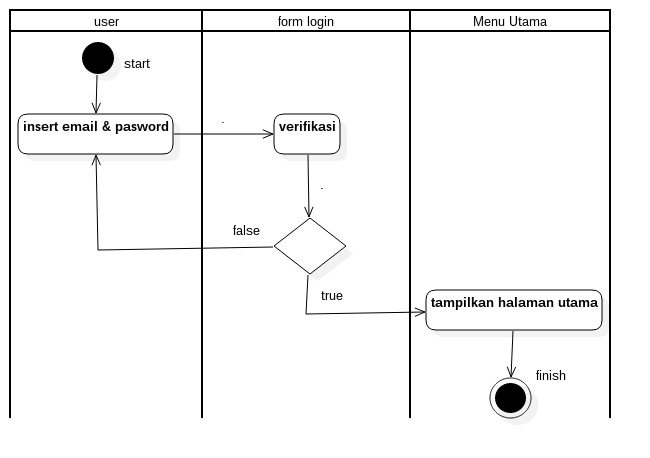
Pada bagian ini admin dapat membuka menu data acara, kemudian memilih acara yang ingin diedit isi datanya. Setelah itu admin dapat mengisikan data baru pada acara yang sudah ada tersebut.



Gambar 3.9. Activity Diagram Edit Data Acara

1. Activity Diagram Login User

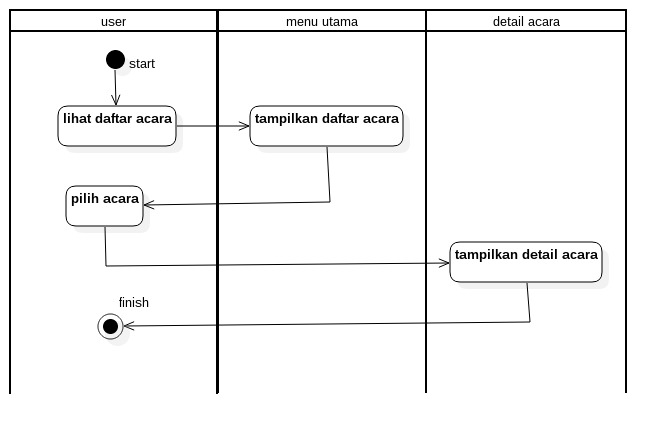
Pada bagian ini *user* memasukkan *email* dan *password* di *form login*, setelah itu *form login* akan melakukan verifikasi data yang dimasukkan tersebut. Apabila *email* atau *password* salah maka akan dilakukan pengulangan untuk memasukkan *email* dan *password* sampai benar.



Gambar 3.10. Activity Diagram Login User

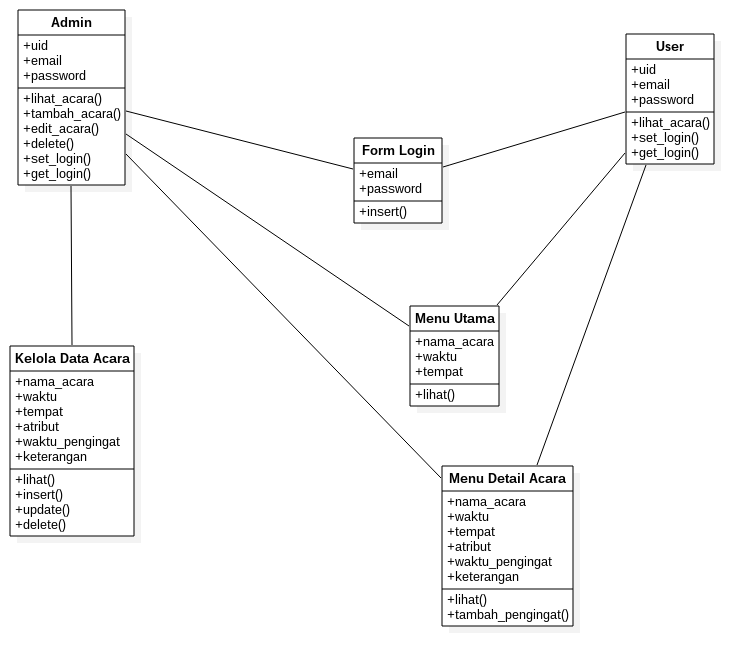
1. Activity Diagram Lihat Acara

Pada bagian ini *user* dapat melihat daftar acara yang ada, setelah itu *user* dapat membuka detail acara tersebut dan melihat detail acara yang ada.



Gambar 3.11. Activity Diagram Lihat Acara

1. **Class Diagram**

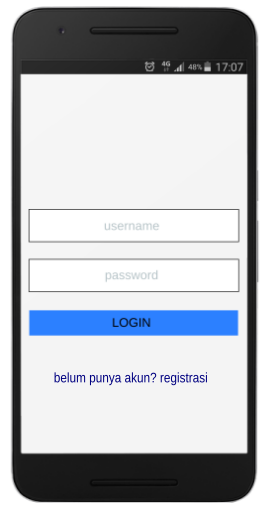


Gambar 3.12. Class Diagram

Pada Gambar 3.12. *class* admin memiliki atribut uid, *email*, dan *password* berasosiasi dengan *class* *form login* yang memiliki atribut *email* dan *password*. *Class* admin juga berasosiasi dengan *class* menu utama yang memiliki atribut nama\_acara, waktu, dan tempat. *Class* admin juga berasosiasi dengan *class* menu detail acara dan *class* kelola data acara yang memiliki atribut nama\_acara, waktu, tempat, atribut, waktu pengingat, dan keterangan. Sedangkan *class user* yang memiliki atribut uid, *email*, dan *password* berasosiasi dengan class form login yang memiliki atribut *email* dan *password*. *Class user* juga berasosiasi dengan *class* menu utama yang memiliki atribut nama\_acara, waktu, dan tempat. *Class* user juga berasosiasi dengan class menu detail acara yang memiliki atribut nama\_acara, waktu, tempat, atribut, waktu pengingat, dan keterangan.

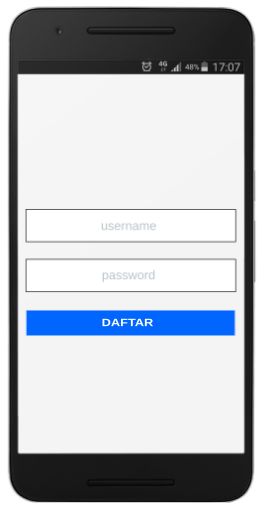
1. **Perancangan Antarmuka**

Untuk konsep perancangan antar muka sistem yang akan dibangun adalah seperti desain berikut ini :



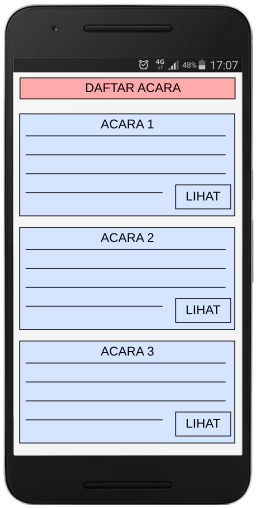
Gambar 3.13. Tampilan Login Aplikasi

Pada Gambar 3.13. adalah tampilan untuk login. Dimana pengguna diminta untuk memasukkan user email berupa email dan password.



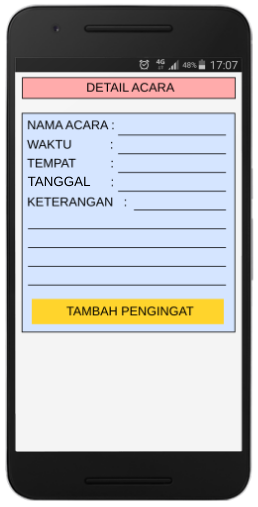
Gambar 3.14. Tampilan Registrasi Untuk Pengguna Aplikasi

Pada Gambar 3.14. adalah tampilan untuk Registrasi. Dimana calon pengguna diminta untuk memasukkan user email berupa email dan password.



Gambar 3.15. Tampilan utama daftar acara

Pada Gambar 3.15. adalah tampilan utama aplikasi yang menunjukkan daftar informasi acara yang ada. Pengguna dapat melihat nama-nama acara yang ada serta dapat membuka acara tersebut melihat informasi lebih lanjut.



Gambar 3.16. Tampilan detail acara

Pada Gambar 3.16. adalah tampilan detail acara yang menampilkan informasi acara berupa nama acara, waktu, tanggal, tempat dan keterangan.



Gambar 3.17. Tampilan admin data acara

Pada Gambar 3.17. adalah tampilan data acara yang menampilkan informasi acara yang ada. Admin dapat menambahkan dapat menambahkan acara baru dengan dengan menyentuh “TAMBAH ACARA BARU”.



Gambar 3.18. Tampilan kelola detail acara

Pada Gambar 3.18. adalah tampilan tambah acara baru. Disini admin harus mengisikan nama acara, waktu, tanggal, tempat, waktu pengingat dan keterangan untuk dapat menambahkan acara baru. Kemudian admin dapat menambahkan acara dengan dengan menyentuh “TAMBAHKAN”.

# **BAB IV**

**IMPLEMENTASI DAN PEMBAHASAN**

* 1. **Implementasi Sistem**

Aplikasi Agenda UKM IK di implementasikan kedalam platform android dengan menggunakan bahasa pemrograman Java. Berdasarkan analisis dan perancangan sistem yang telah dibuat di bab 3 sebelumnya, maka implementasi merupakan tahap dimana sistem siap dioperasikan dan di uji pada keadaan yang sebenarnya. Berikut implementasi sistem yang ada :

|  |
| --- |
| private void addClassified(ClassifiedEvent classifiedEvent, String cEventId) {  classifiedEvent.setEventId(cEventId);  dbRef.child("classified").child(cEventId)  .setValue(classifiedEvent)  .addOnCompleteListener(newOnCompleteListener  <Void>() {    @Override  public void onComplete(@NonNull Task<Void> task) {  if (task.isSuccessful()) {  if(isEdit){  addClassifieds();  }else{  restUi();  }  Log.d(TAG, "Classified berhasil di tambahkan  ke database");  Toast.makeText(getActivity(),"Acara berhasil  di posting",Toast.LENGTH\_SHORT).show();  }  }  private ClassifiedEvent createClassifiedEventObj() {  final ClassifiedEvent event = new ClassifiedEvent();  event.setNama(((EditText) getActivity()  .findViewById(R.id.nama\_a)).getText().toString());  event.setKategori(((EditText) getActivity()  .findViewById(R.id.category\_a)).getText().toString());  event.setDeskripsi(((EditText) getActivity()  .findViewById(R.id.desc\_a)).getText().toString());  event.setTanggal(((EditText) getActivity()  .findViewById(R.id.tanggal\_a)).getText().toString());  event.setWaktu(((EditText) getActivity()  .findViewById(R.id.waktu\_a)).getText().toString());  event.setNotif(((EditText) getActivity()  .findViewById(R.id.notif\_a)).getText().toString());  event.setTempat(((EditText) getActivity()  .findViewById(R.id.tempat\_a)).getText().toString());  return event;  } |

Gambar 4.1. Menambahkan data acara

Pada Gambar 4.1. merupakan potongan *source code* dari aplikasi bagian admin yang digunakan untuk menambahkan data acara. Dimana data acara yang ada meliputi nama acara, kategori, deskripsi, tanggal, waktu, waktu pengingat notifikasi, dan tempat nantinya akan di posting langsung ke *Firebase Realtime Database* dengan nama “classified”

|  |
| --- |
| private void getClassifiedsFromDb(final String kategori) {  databaseReference.child("classified").orderByChild("kategori") .equalTo(kategori).addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(DataSnapshot dataSnapshot) {  List<ClassifiedEvent> eventsList = new ArrayList<ClassifiedEvent>();  for (DataSnapshot eventSnapshot: dataSnapshot.getChildren()) { eventsList.add(eventSnapshot.getValue(ClassifiedEvent.class));}  Log.d(TAG, "no of events for search is "+eventsList.size());  EventRecyclerView recyclerViewAdapter = new EventRecyclerView(eventsList, getActivity());  eventsRecyclerView.setAdapter(recyclerViewAdapter);  }  @Override  public void onCancelled(DatabaseError databaseError) {  Log.d(TAG, "Error trying to get classified events for " +kategori+" "+databaseError);  Toast.makeText(getActivity(),  "Error trying to get classified events for " +kategori, Toast.LENGTH\_SHORT).show(); }  });} |

Gambar 4.2. Melihat daftar acara

Pada Gambar 4.2. merupakan potongan *source code* dari aplikasi bagian admin yang digunakan untuk melihat daftar acara. Daftar acara yang akan ditampilkan disini diambil dengan memasukkan nilai pada data “kategori” yang dijadikan sebagai parameter kemudian akan ditampilkan daftar acara yang ada kedalam *Recycler View.*

|  |
| --- |
| private void sendNotification() {  JSONObject json = new JSONObject(); try { json.put("to","/topics/"+"news"); JSONObject notificationObj = new JSONObject();  notificationObj.put("title","UKM IK EVENT");  notificationObj.put("body","ada acara baru nih guys, ayo silahkan dicek !");  json.put("notification",notificationObj);  JsonObjectRequest request = new JsonObjectRequest(Request.Method.POST, URL, json, new Response.Listener<JSONObject>() {  @Override  public void onResponse(JSONObject response) {  Log.d("MUR", "onResponse: "); }}, new Response.ErrorListener() {  @Override  public void onErrorResponse(VolleyError error) {  Log.d("MUR", "onError: "+error.networkResponse);}  })  {  @Override  public Map<String, String> getHeaders() throws AuthFailureError { Map<String,String> header = new HashMap<>(); header.put("content-type","application/json"); header.put("authorization","key=AIzaSyB1C7zEeYTlRI52w2lhcxl\_JGuebql9S3Q");  return header; }  }; |

Gambar 4.3. Mengirim push notification

Pada Gambar 4.3. merupakan potongan *source code* dari aplikasi bagian admin yang digunakan untuk mengirim push notification pada saat ada acara baru yang telah ditambahkan. Target notifikasi yang dikirimkan disini dapat dilihat pada bagian "authorization", "key=AIzaSyB1C7zEeYTlRI52w2lhcxl\_JGuebql9S3Q". Dimana key tersebut merupakan kode unik dari *project* aplikasi ini yang telah diintegrasikan kedalam *Firebase Console*. Sedangkan broadcast notifikasi disini menggunakan metode topic yang disini diberi nama “news”. Kemudian judul notifikasi diset “UKM IK EVENT”, sedangkan untuk isi dari notifikasi tersebut di set “ada acara baru nih guys, ayo silahkan dicek !”.

|  |
| --- |
| private void populateUpdateEvent() {  headTxt.setText("Edit Acara");  button.setText("Edit Acara");  isEdit = true;  dbRef.child("classified").child(eventId).  addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(DataSnapshot dataSnapshot) {  ClassifiedEvent cEvent = dataSnapshot.getValue(ClassifiedEvent.class);  displayEventForUpdate(cEvent);  }  @Override  public void onCancelled(DatabaseError databaseError) { Log.d(TAG, "Error trying to get classified event for update "+""+databaseError);  Toast.makeText(getActivity(),  "Please try classified edit action again", Toast.LENGTH\_SHORT).show();  }  }); |

Gambar 4.4. Mengedit data acara

Pada Gambar 4.4. merupakan potongan *source code* dari aplikasi bagian admin yang digunakan untuk mengedit data acara yang sudah ada. Pada bagian onDatachange merupakan fungsi yang bertanggung jawab untuk menyimpan perubahan yang ada pada data yang telah diedit tersebut.

|  |
| --- |
| private void deleteClassifiedEvent(String eventId, final int position){  FirebaseDatabase.getInstance().getReference() .child("classified").child(eventId).removeValue()  .addOnCompleteListener(new OnCompleteListener <Void>() {    @Override  public void onComplete(@NonNull Task<Void> task) {  if (task.isSuccessful()) { eventsList.remove(position);  notifyItemRemoved(position); notifyItemRangeChanged(position, eventsList.  size());    Log.d("Delete Event", "Classified has been deleted");  Toast.makeText(context,"Classified has been  deleted", Toast.LENGTH\_SHORT).show();  } else { Log.d("Delete Event", "Classified couldn't be deleted");  Toast.makeText(context,"Classified could not be deleted", Toast.LENGTH\_SHORT).show(); }}  });  } |

Gambar 4.5. Menghapus data acara

Pada Gambar 4.5. merupakan potongan *source code* dari aplikasi bagian admin yang digunakan untuk menghapus data acara yang ada. eventId merupakan variable yang digunakan untuk menginisiasi id dari masing-masing acara yang ada. Kemudia acara yang dipilih untuk dihapus tersebut dikenali berdasarkan id nya dan sintak removeValue() yang akan bertugas untuk menhapus acara tersebut dengan diteruskan ke fungsi onComplete untuk memproses lebih lanjut penghapusan.

|  |
| --- |
| private void registerUser() {  String email = editTextEmail.getText().toString().  trim();  String password = editTextPassword.getText().toString().  trim();  mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener <AuthResult>()  } |

Gambar 4.6. Registrasi

Pada Gambar 4.6. merupakan potongan *source code* dari aplikasi bagian pengguna yang digunakan untuk melakukan registrasi. Dimana pada fungsi registerUser() terdapat komponen string dari “email” dan “password” yang telah diambil dari editTextEmail dan editTextPassword yang telah dimasukkan oleh pengguna pada tampilan regsitrasi. Kemudian sintak createUserWithEmailAnd Password adalah yang bertugas untuk mendaftarkan email dan password yang telah dimasukkan oleh pengguna tersebut.

|  |
| --- |
| private void userLogin() {  String email = editTextEmail.getText().toString().  trim();  String password = editTextPassword.getText().  toString().trim();  mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>()  } |

Gambar 4.7. Login

Pada Gambar 4.7. merupakan potongan *source code* dari aplikasi bagian pengguna yang digunakan untuk melakukan login pada aplikasi. Dimana pada fungsi userLogin() terdapat komponen string dari “email” dan “password” yang telah diambil dari editTextEmail dan editTextPassword yang telah dimasukkan oleh pengguna pada tampilan login. Kemudian sintak signInWithEmailAndPassword adalah yang bertugas untuk melakukan login pada *Firebase Authentication* agar bisa masuk ke halaman utama aplikasi.

|  |
| --- |
| public ArrayList<Classified> retrieve() { db.child("classified").addValueEventListener(new ValueEventListener() {    @Override  public void onDataChange(DataSnapshot dataSnapshot) {  classifieds.clear();  if (dataSnapshot.exists() && dataSnapshot.getChildrenCount() > 0) {  for (DataSnapshot ds : dataSnapshot.getChildren()) {  Classified classified = ds.getValue(Classified.class); classifieds.add(classified); }  adapter = new CustomAdapter(c, classifieds);  mListView.setAdapter(adapter);  new Handler().post(new Runnable() {  }} |

Gambar 4.8. Mengambil data dari *Firebase Relatime Database*

Pada Gambar 4.8. merupakan potongan *source code* dari aplikasi bagian pengguna yang digunakan untuk mengambil data acara yang ada pada *Firebase Realtime Database*. Sintak retrieve() merupakan kunci dimana data acara yang ada pada *Firebae Realtime Database* teresebut dapat diambil melalui aplikasi ini.

|  |
| --- |
| public View getView(int position, View convertView, ViewGroup parent) {  if (convertView == null) { convertView = LayoutInflater.from(c).inflate(R.layout.model, parent, false);  }  TextView nameTextView = convertView.findViewById  (R.id.nameTextView);  TextView quoteTextView = convertView.findViewById  (R.id.quoteTextView);  TextView descriptionTextView = convertView.  findViewById(R.id.descriptionTextView);  final Classified s = (Classified)this.  getItem(position);  nameTextView.setText(s.getNama());  quoteTextView.setText(s.getDeskripsi());  descriptionTextView.setText(s.getTanggal()); |

Gambar 4.9. Menampilkan data acara pada halaman utama aplikasi

Pada Gambar 4.9. merupakan potongan *source code* dari aplikasi bagian pengguna yang digunakan untuk menampilkan data yang sudah diambil sebelumnya kedalam listview halaman utama, dimana pada halaman ini pengguna nantinya dapat melihat daftar acara yang ada meliputi nama acara, deskripsi acara, dan tanggal acara.

|  |
| --- |
| protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_detail);  namaTxt = (TextView) findViewById(R.id.namaDetailTextView);  tanggalTxt = (TextView) findViewById(R.id.tanggalDetailTextView);  waktuTxt = (TextView) findViewById(R.id.waktuDetailTextView);  tempatTxt = (TextView) findViewById(R.id.tempatDetailTextView);  deskripsiTxt = (TextView) findViewById(R.id.deskripsiDetailTextView);  notifTxt = (TextView) findViewById(R.id.notifTxt);  alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);    //GET INTENT  Intent i = this.getIntent();  //TERIMA DATA DARI HALAMAN UTAMA  String nama = i.getStringExtra("NAMA\_KEY");  String tanggal = i.getStringExtra("TANGGAL\_KEY");  String waktu = i.getStringExtra("WAKTU\_KEY");  String tempat = i.getStringExtra("TEMPAT\_KEY");  String deskripsi = i.getStringExtra("DESKRIPSI\_KEY");  String notif = i.getStringExtra("NOTIF\_KEY");  //TAMPILKAN DATA PADA HALAMAN DETAIL  namaTxt.setText(nama);  tanggalTxt.setText(tanggal);  waktuTxt.setText(waktu);  tempatTxt.setText(tempat);  deskripsiTxt.setText(deskripsi);  notifTxt.setText(notif);  } |

Gambar 4.10. Halaman detail acara

Pada Gambar 4.10. merupakan potongan *source code* dari aplikasi bagian pengguna yang digunakan untuk menampilkan detail dari data acara yang dipilih oleh pengguna. Dimana pada halaman detail acara ini sintak String nama = i.getStringExtra("NAMA\_KEY"); dan dibawahnya berfungsi untuk mendapatkan data yang telah dikirim melalui halaman utama dari salah satu acara yang telah dipilih oleh pengguna. Kemudian sintak namaTxt.setText(nama); berfungsi untuk menampilkan data detail acara tersebut kedalam halaman Detail Acara.

|  |
| --- |
| public void onMessageReceived(RemoteMessage remoteMessage) {  super.onMessageReceived(remoteMessage);  if(remoteMessage.getData().isEmpty())  showNotification(remoteMessage.getNotification().  getTitle(),remoteMessage.getNotification().  getBody());  else  showNotification(remoteMessage.getData());  } |

Gambar 4.11. Pengampu push notification

Pada Gambar 4.11. merupakan potongan *source code* dari aplikasi bagian pengguna yang digunakan untuk mengampu push notification yang masuk. Dimana fungsi onMessageReceived merupakan kunci utamanya. Kemudian sintax showNotification berfungsi untuk menampilkan notifikasi yang masuk ke dalam layar *home tile* smarphone pengguna aplikasi Agenda UKM IK.

|  |
| --- |
| public void ingatkanSaya(View view) {  String date = Converter.ConvertDate(tanggalTxt.getText().toString(), "dd/MM/yyyy", "yyyy MM dd");    String time = notifTxt.getText().toString();  String dateTime = date + " " + time;  Date txtAlarm =  Converter.toDate(dateTime, "yyyy MM dd HH:mm");  Log.d("MyActivity", "Alarm On");  Calendar calendar = Calendar.getInstance();  calendar.setTime(txtAlarm);  Intent myIntent = new Intent(this,  AlarmReciever.class);  pendingIntent = PendingIntent.getBroadcast(DetailActivity.this, 0, myIntent, 0);  alarmManager.set(AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), pendingIntent);  Toast.makeText(getApplicationContext(), "Pengingat telah dibuat !", Toast.LENGTH\_SHORT).show();  } |

Gambar 4.12. Menambahkan pengingat acara

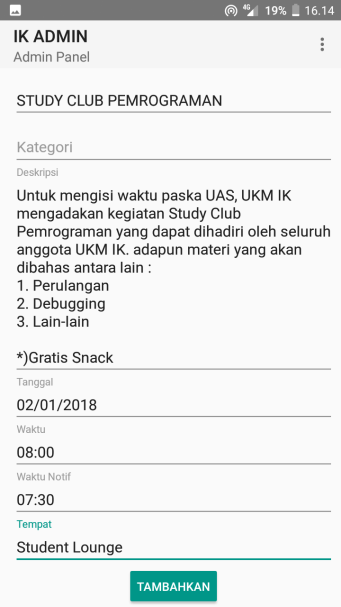
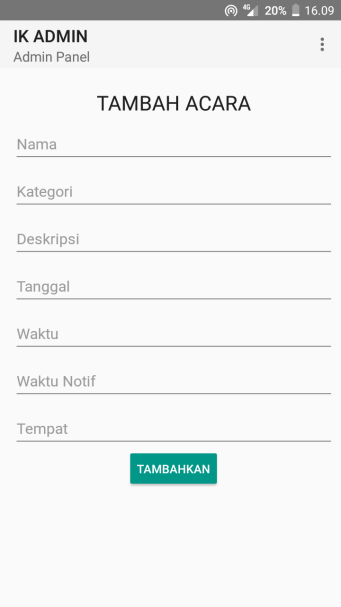
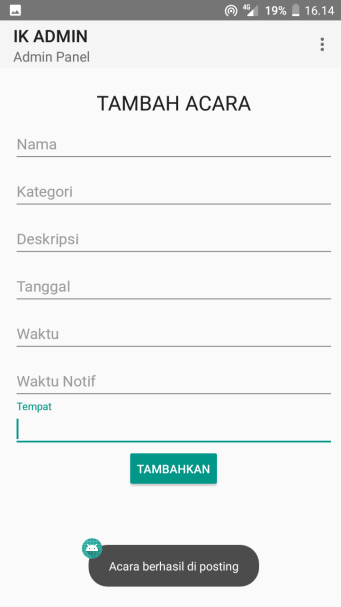
Pada Gambar 4.12. merupakan potongan *source code* dari aplikasi bagian pengguna yang digunakan untuk untuk menambahkan pengingat acara. Dimana hal pertama yang dilakukan oleh sistem adalah mendapatkan nilai tanggal yang ada berupa “dd/mm/yyy” kemudian nilai tanggal tersebut dilakukan konversi kedalam format sistem berupa “yy mm dd”. Kemudian untuk waktu diambil dari nilai variabel “notifTxt” yang tidak perlu dikonversi lagi. Selanjutnya nilai sintak String time = merupakan penggabungan antara tanggal dan waktu diatas untuk dijadikan waktu pengingat acara. Dan terakir sintak alarmManager.set (AlarmManager.RTC\_WAKEUP, calendar.getTimeInMillis(), pendingIntent); merupakan sintak yang bertugas untuk membuat pengigat acara.

* 1. **Uji Coba dan Pembahasan**

Pada bagian ini akan membahas mengenai hasil dari implementasi sistem yang telah di uji coba dengan dijalankan di perangkat smartphone android dengan keadaan yang sebenarnya.

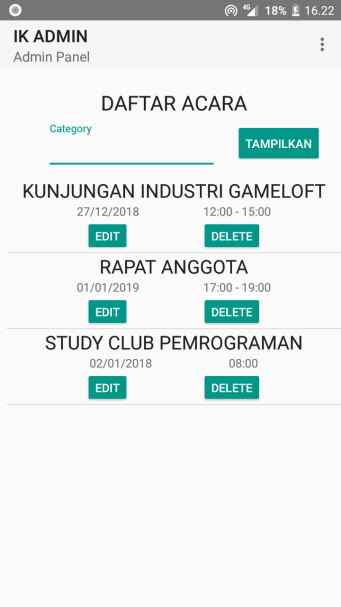
* + 1. **Menambahkan Acara Baru Dari Aplikasi Admin**

Halaman ini merupakan halaman yang digunakan oleh admin untuk menambahkan acara baru. Data yang ditambahkan tersebut nantinya akan disimpa langsug ke Firebase Realtime Database dan sistem akan mengirim push notification secara otomatis kepada pengguna aplikasi Agenda UKM IK bahwa ada acara baru yang ditambahan.

****

Gambar 4.13. Menambahkan Acara

Pada Gambar 4.13. diatas admin dapat menambahkan acara baru meliputi atribut nama, kategori, deskripsi, tanggal, waktu, waktu notif, dan tempat. Kemudian saat tombol “TAMBAHKAN” diklik maka akan muncul Toast “Acara berhasil diposting” yang menandakan acara tersebut berhasil ditambahkan. Untuk mengecek apakah acara ini sudah benar-benar masuk ke database. Maka admin dapat beralih melihat daftar acara yang ada seperti gambar dibawah ini :

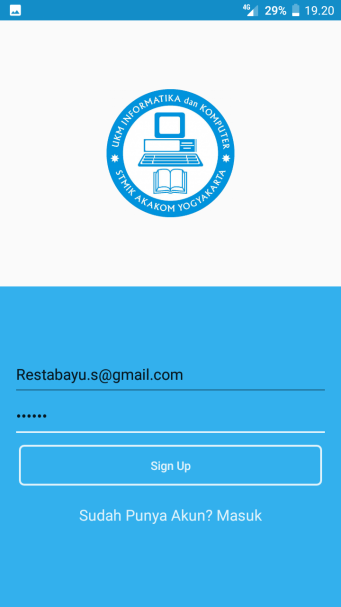
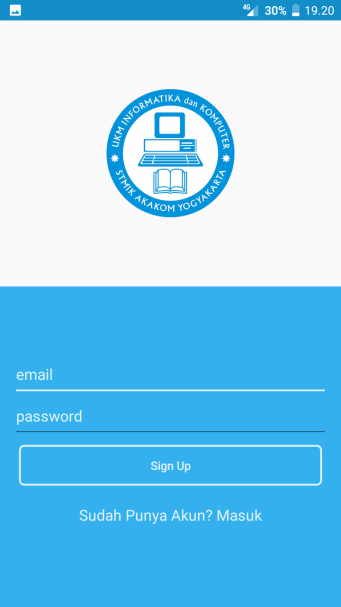
****

Gambar 4.14. Daftar acara yang ada

Dari Gambar 4.14. diatas terlihat bahwa acara yang ditambahkan oleh admin sebelumnya sudah masuk berada dikolom paling bawah yaitu “STUDY CLUB PEMROGRAMAN”.

* + 1. **Halaman Registrasi Aplikasi Pengguna**

Halaman registrasi pengguna merupakan halaman untuk membuat akun bagi anggota UKM IK selaku pengguna aplikasi ini. Data dari pengguna yang berhasil dibuat kemudian akan disimpan ke server firebase authentication.

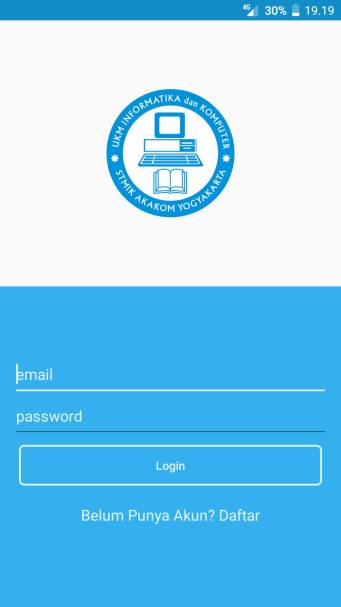
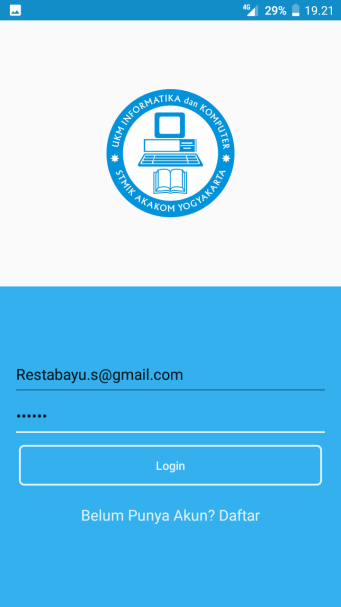


Gambar 4.15. Halaman Registrasi aplikasi Pengguna

Pada Gambar 4.15. diatas terdapat form registrasi berupa email dan password. Dimana calon pengguna disini diminta untuk memasukkan alamat email beserta password untuk dapat mendaftarkan diri sebagai pengguna aplikasi ini.

* + 1. **Halaman Login Aplikasi Pengguna**

Halaman login pengguna merupakan halaman untuk melakukan login ke aplikasi ini bagi anggota UKM IK selaku pengguna yang sudah memiliki akun sebelumnya.

Gambar 4.16. Halaman Login aplikasi Pengguna

Pada Gambar 4.16. diatas terdapat form login berupa email dan password. Dimana pengguna disini diminta untuk memasukkan alamat email beserta password yang sudah terdaftar untuk dapat melakukan login ke aplikasi ini. Saat tombol login di klik maka sistem akan melakukan verifikasi antara data yang telah dimasukkan oleh pengguna dengan data yang ada pada server firebase authentication apakah cocok atau tidak, apabila tidak cocok maka sistem akan meminta pengguna untuk memasukkan data yang valid. Hingga setelah data yang dimasukkan cocok maka sistem akan membuka halaman Slide Show seperti dibawah ini yang menandakan bahwa pengguna berhasil login.



Gambar 4.17. Halaman Slide Show aplikasi Pengguna

Pada Gambar 4.17. diatas merupakan Slide Show yang menyambut pengguna karena telah berhasil login ke aplikasi ini. Untuk slide pertama menampilkan cerita singkat tentang UKM IK dan sejarahnya, sedangkan untuk slide kedua menampilkan fitur dari aplikasi ini.

* + 1. **Halaman Utama Aplikasi Pengguna**

Halaman utama aplikasi pengguna merupakan halaman yang menampilkan daftar keseluruhan acara yang ada. Kemudian pada halaman ini juga terdapat menu toolbar yang berisi “Pengurus” untuk melihat daftar pengurus UKM IK yang ada, “Help” untuk melihat petunjuk cara penggunaan aplikasi dan “Logout” untuk keluar dari akun aplikasi ini.

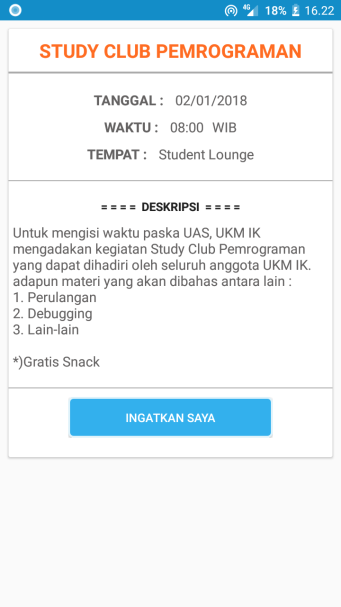


Gambar 4.18. Halaman Utama aplikasi Pengguna

Pada Gambar 4.18. diatas merupakan halaman utama aplikasi pengguna. Dimana pada halaman tersebut terlihat ada beberapa agenda acara. Dimana acara yang paling baru termuat di urutan paling atas seperti yang dapat dilihat yaitu “STUDY CLUB PEMROGRAMAN”.

* + 1. **Halaman Detail Acara Aplikasi Pengguna**

Halaman detail acara aplikasi pengguna merupakan halaman yang menampilkan detail acara yang ada meliputi nama acara, tanggal, waktu, tempat dan juga deskripsi acara yang ada. Kemudian disini terdapat tombol “INGATKAN SAYA” dimana apabila tombol ini ditekan maka akan membuat pengingat secara otomatis akan memunculkan notifikasi pada saat acara yang ada akan segera dimulai.

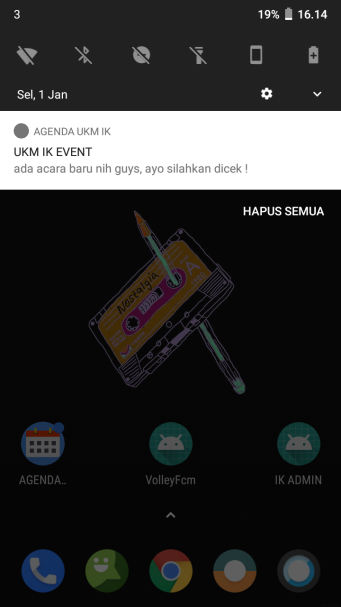
****

Gambar 4.19. Halaman Detail Acara aplikasi Pengguna

Pada Gambar 4.19. diatas merupakan halaman detail acara, dimana jika diperhatikan terdapat beberapa komponen yang berisi “STUDY CLUB PEMROGRAMAN” merupakan nama acara, kemudian terdapat tanggal, waktu, tempat dan juga deskripsi pada acara tersebut. Pada bagian paling bawah terdapat tombol “INGATKAN SAYA” dimana apabila tombol ini diklik maka sistem akan membuat pengingat notifikasi yang secara otomatis akan muncul saat acara tersebut akan dimulai.

* + 1. **Notifikasi Acara Baru**

Notifikasi acara baru merupakan push notification yang didapat oleh pengguna aplikasi Agenda UKM IK apabila ada acara baru yang ditambahkan oleh admin.

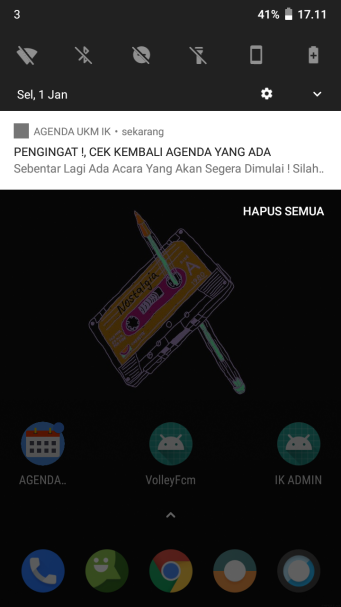


Gambar 4.20. Notifikasi Acara Baru

Pada gambar 4.20. diatas terlihat muncul sebuah notifikasi dengan judul “UKM IK EVENT” dan pesan “ada acara baru nih guys, ayo silahkan dicek !” dimana notifikasi tersebut mengartikan baha ada acara baru yang ditambahkan pada aplikasi Agenda UKM IK. Maka pengguna dapat membuka aplikasi tersebut untuk melihat acara baru tersebut.

* + 1. **Notifikasi Pengingat Acara**

Notifikasi Pengingat acara merupakan notifikasi yang didapat oleh pengguna aplikasi Agenda UKM IK secara otomatis apabila ada acara yang akan segera dimulai.

****

Gambar 4.21. Notifikasi Pengingat Acara

Pada gambar 4.21. diatas terlihat muncul sebuah notifikasi dengan judul “PENGINGAT !, CEK KEMBALI AGENDA YANG ADA” dan pesan “sebentar lagi ada acara yang akan segera dimulai !” dimana notifikasi tersebut mengartikan baha ada salah satu acara UKM IK yang akan segera dimulai. Maka pengguna dapat membuka aplikasi tersebut untuk melihat acara yang akan dimulai tersebut agar tidak ketinggalan.

# **BAB V**

# **KESIMPULAN DAN SARAN**

**5.1. Kesimpulan**

Berdasarkan penelitian yang dilakukan melalui perancangan yang dilakukan pada bab-bab sebelumnya, maka dapat diambil kesimpulan sebagai beikut :

* + - 1. Berhasil membuat aplikasi Agenda UKM IK berbasis android dengan penerapan tenologi Firebase Authentication, Firebase Cloud Messaging dan Firebase Realtime Database.
      2. Admin dapat mengirim acara secara serentak kepada seluruh pengguna aplikasi Agenda UKM IK.
      3. Pengguna Aplikasi Agenda UKM IK dapat menerima notifikasi acara saat ada acara baru yang ditambahkan.
      4. Pengguna Aplikasi Agenda UKM IK dapat menambahkan pengingat pada acara yang ada agar tidak ketinggalan saat acara akan dimulai.
  1. **Saran**

Bedasarkan hasil penelitian dan implementasi teknologi *firebase* kedalam aplikasi android dalam penelitian ini masih terdapat beberapa kekurangan. Sehingga untuk untuk membuat aplikasi ini menjadi lebih baik lagi maka diberikan saran seperti berikut :

Perlunya dilakukan optimasi untuk bagian *user experience* aplikasi agar notifikasi yang masuk saat dibuka oleh pengguna dapat men trigger langsung ke detail acara yang ada.

Perlunya penambahan fitur chatting agar pengguna dapat saling berinteraksi melalui aplikasi ini.

# **DAFTAR PUSTAKA**

Aghnia Fi’la Urfan. 2017. *“Aplikasi Kalender Event Seni Kontemporer di DIY*

*Dengan Menggunakan Notifikasi Google Calender dan Pengarahan Lokasi Menggunakan Google Map Berbasis Android”*. Stmik Akakom Yogyakarta. Yogyakarta.

Belal Khan. 2016. *“Firebase Cloud Messaging Tutorial for Android”*.

<https://www.simplifiedcoding.net/firebase-cloud-messaging-tutorial-android/.> Diakses pada tanggal 1 April 2018. 12:51 WIB.

Google Develper Training Team. 2016. *“Android Developer Fundamentals*

*Course”*.

[https://chromplex.com/unduh-ebook-pemrograman-android-berbaha](https://www.simplifiedcoding.net/firebase-cloud-messaging-tutorial-android/.)

[sa-indonesia-resmi-dari-google/.](https://www.simplifiedcoding.net/firebase-cloud-messaging-tutorial-android/.) Diakses pada tanggal 23 Maret

2018. 05:13 WIB.

Mursito. 2017. *“Penggunaan Firebase Cloud Messaging pada Aplikasi Log*

*book TAS”*. Stmik Akakom Yogyakarta. Yogyakarta.

Reyhan Pradipta Sumardi. 2017. *“Aplikasi Mobile Notification Informasi*

*Perkuliahan Berbasis Android”*. Stmik Akakom Yogyakarta.

Yogyakarta.

Reza Abdillah. 2017. *“Implementasi Teknologi Push Notification pada*

*Aplikasi Lowongan Kerja Berbasis Android”*. Stmik Akakom

Yogyakarta. Yogyakarta.

Silva Casiavera. 2018. *“Aplikasi Pengingat Imunisasi Pada Anak Berbasis*

*Android”*. Stmik Akakom Yogyakarta. Yogyakarta.

Statista. 2018. “*Number of available applications in the Google Play Store from December 2009 to June 2018*".

[https://www.statista.com/statistics/266210/number-of-available-appl](https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/)

[ications-in-the-google-play-st/](https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/). Diakses pada tanggal 4 juli 2018.

17:07 WIB.