

**PEMBANGUNAN SISTEM INFORMASI *UNTIRTA ONLINE*
TEST (UNOT) BERBASIS *ANDROID***

LAPORAN HASIL PENELITIAN

Diajukan untuk memenuhi syarat kelulusan Mata Kuliah Aplikasi Bergerak

Dosen Pengampu : Fadil Muhammad, S.T., M.T.



Disusun Oleh :

GALIH AJI PAMBUDI (3332180058)

DIMAS EMERALDO ASYDIQI (3332180065)

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS SULTAN AGENG TIRTAYASA
2021**

PRAKATA

Puji dan Syukur dipanjatkan kepada Allah SWT yang telah memberikan nikmat dan karunia yang sangat banyak sehingga, kami dapat menyelesaikan *project* aplikasi bergerak dengan baik serta dapat *menuliskan* hasil yang telah didapatkan selama kegiatan tersebut.

Kami membuat Sistem Informasi *Test* Berbasis *Android* yang Bernama UNOT selain sebagai *pemenuh* tugas mata kuliah aplikasi bergerak juga dipengaruhi dari pandemi COVID-19 yang menyebabkan ketidakpastian dalam melakukan kegiatan belajar mengajar.

Kami berterimakasih kepada seluruh pihak yang telah mendukung kami selama mengerjakan project ini. Kami mohon maaf kepada seluruh pihak yang telah direpotkan, kekurangan pada laporan ini, dan berbagai kekurangan pada sistem informasi yang telah dibuat sehingga, kami sangat senang apabila pembaca mau memberikan kritik dan sarannya terhadap laporan ini. Akhir kata, kami harap semoga laporan ini dapat menginspirasi dan dapat menjadi referensi untuk penelitian selanjutnya.

Serang, 20 Desember 2021

Galih AP / Dimas EA

DAFTAR ISI

PRAKATA.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL.....	iv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Tujuan Penelitian.....	1
1.3. Manfaat Penelitian.....	2
1.4. Ruang Lingkup Penelitian	2
1.5. Sistematika Penulisan.....	2
BAB II TINJAUAN PUSTAKA.....	4
BAB III LANDASAN TEORI.....	6
3.1. Sejarah Sistem Operasi <i>Android</i>	6
3.2. Pemrograman Aplikasi <i>Android</i>	9
BAB IV PEMBAHASAN.....	17
4.1. Alur Kegiatan.....	17
4.2. Struktur Aplikasi.....	20
BAB V PENUTUP.....	26
5.1. Kesimpulan	26
5.2. Saran dan Kritik	27
DAFTAR PUSTAKA	28
LAMPIRAN.....	30

DAFTAR GAMBAR

Gambar 3.1 Sistem Operasi <i>Android</i> 11	7
Gambar 3.2 Arsitektur <i>Android</i> (Murphi, <i>Beginning Android</i> , 2009)	9
Gambar 3.3 Logo <i>Startup Android Studio IDE</i>	9
Gambar 3.4 <i>GUI Android Studio</i>	10
Gambar 3.5 Logo Bahasa <i>Java</i>	11
Gamabr 3.6 <i>Toolchain Java</i>	12
Gambar 3.7 Hierarki <i>Layout</i> Dalam Mendefinisikan UI	13
Gambar 3.8 Mekanisme <i>Recycler View</i> Bekerja.....	16
Gambar 4.1 Flowchart UNOT.....	17
Gambar 4.2 Splash, Login User, dan Login Admin.....	20
Gambar 4.3 <i>Dashboard Admin, CRUD Soal, Kontrol Aplikasi</i>	21
Gambar 4.4 Nilai Peserta, Absen Peserta, dan Dashboard Peserta	22
Gambar 4.5 Info Akun, <i>Test Activity</i> , dan Hasil <i>Activity</i>	23
Gambar 4.6 <i>Webview Siakad, Feed, Save pdf, Scan QR Code</i>	24
Gambar 4.7 <i>Firebase Database</i>	25

DAFTAR TABEL

Tabel 3.1 Daftar Versi <i>Android</i> dari 1.5 – 10	7
---	---

BAB I

PENDAHULUAN

1.1. Latar Belakang

Hampir tiga tahun sejak Indonesia dilanda pandemi *COVID-19*, sejak saat itu pula masyarakat Indonesia membatasi semua kegiatan yang dilakukan dengan menerapkan *physical distancing*. Akibatnya aktivitas yang semula dilakukan berkelompok ataupun berada pada satu ruangan bersama, kini dilakukan secara *online*. Hal ini pun berdampak pada kegiatan belajar mengajar, ujian, dan kegiatan penelitian mahasiswa pun harus dilakukan secara *online*. Akibatnya, mahasiswa pun dirugikan karena ilmu yang mereka serap tidak lagi maksimal. Sehingga perlu adanya suatu sistem informasi yang memberikan akses pelayanan maupun fungsi yang sesuai dengan kebutuhan mahasiswa baik itu sistem informasi *administrasi* ataupun sistem informasi untuk melakukan ujian *online*.

Sistem informasi tes ini dibuat dalam bentuk aplikasi *Android* yang bersifat *mobile* sehingga mudah dibawa kemanapun dan dapat dilakukan dimanapun. Perangkat yang digunakan untuk *Android* juga memiliki fitur yang setara dengan perangkat *PC*, bahkan hampir semua mahasiswa memiliki perangkat ini. Sehingga dengan melihat kepopuleran serta keandalan *Android*, maka sangat mungkin kita dapat membuat sistem informasi kompleks yang memanfaatkan fitur sensor pada perangkat.

1.2. Tujuan Penelitian

Laporan ini memiliki tujuan umum untuk menambah wawasan mahasiswa terhadap project berbasis *Android* dan memiliki tujuan utama yaitu :

- a. Membangun aplikasi *Untirta Online Test (UNOT)* berdasarkan kegiatan proyek secara langsung.
- b. Menerapkan penggunaan Bahasa *Java 8* dan Bahasa *Markup XML* pada proses pembuatan aplikasi *UNOT*.

- c. Memanfaatkan penggunaan salah satu sensor pada perangkat dan memanfaatkan *realtime database* dari google (*Firestore Database*)
- d. Melakukan uji coba kelayakan pada aplikasi *UNOT*.

1.3. Manfaat Penelitian

Diharapkan dari kegiatan berbasis project pada mata kuliah aplikasi bergerak ini adalah :

- a. Aplikasi dapat bermanfaat baik untuk digunakan ataupun untuk penelitian selanjutnya.
- b. Aplikasi dapat digunakan sebagai sistem informasi *test* di UNTIRTA.
- c. Aplikasi dapat menjadi motivasi orang untuk beralih dari konvensional ke digital.

1.4. Ruang Lingkup Penelitian

Ruang lingkup dikemukakan pada bagian variabel-variabel yang akan diteliti, subjek penelitian, lokasi penelitian, serta batasan-batasan lain yang dipakai pada penelitian ini, yaitu :

- a. Pengembangan aplikasi harus dilakukan secara *WFH (Work From Home)* dengan memanfaatkan github sebagai *online repository*.
- b. Pengembangan yang dilakukan hanya ditujukan untuk pengguna perangkat *Android*.
- c. Aplikasi yang dihasilkan harus berjalan dengan baik serta bersifat dinamis dan *Open Source* sehingga dimasa mendatang, prototype ini dapat dikembangkan kembali.

1.5. Sistematika Penulisan

Laporan ini disusun menjadi 5 bagian utama, yaitu Pendahuluan yang berisi tentang informasi umum dari penelitian, seperti latar belakang, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian, dan sistematika penulisan. Kemudian, Tinjauan Pustaka yang berisi *State Of The Art* beberapa penelitian terdahulu dengan tema atau pembahasan yang mirip dengan tema yang diangkat dalam laporan. Lalu,

Landasan Teori yang berisi teori-teori/prinsip kerja yang telah dibahas oleh ahli atau peneliti sebelumnya. Kemudian, Pembahasan yang berisi tentang isi utama berupa hasil yang diperoleh dari kerja praktik serta dalam bentuk penjabaran langsung oleh penulis. Terakhir, Penutup yang berisi kesimpulan dari laporan kerja praktik secara keseluruhan serta kritik dan saran dari laporan sebagai evaluasi yang dapat diambil hikmahnya oleh penulis maupun pembaca lainnya.

BAB II

TINJAUAN PUSTAKA

Berbagai penelitian yang mengkaji mengenai sistem informasi berbasis *Android* maupun sistem informasi tes *TOEFL* sudah banyak dilakukan sebelumnya. Jurnal maupun teori yang membahas kemiripan teori maupun subjek penelitian digunakan untuk memperkuat teori serta judul yang saya angkat sebagai laporan kerja praktik ini. Berikut merupakan beberapa penelitian terdahulu yang pernah dilakukan sebelumnya sebagai referensi laporan ini.

Pada tahun 2013, Alvin Assianto Leiman dkk dalam tesisnya yang berjudul *Aplikasi Ujian Online Pada Mobile Device Android* membuat sistem informasi ujian online berbasis *android* menggunakan *Web Service with SOAP* serta Bahasa yang digunakan yaitu *Jquery*, *PHP*, dan *XML*. Hasil yang didapat berupa *web* sistem yang dapat diakses melalui *browser* maupun aplikasi *android* yang telah dibuat lengkap dengan *login process* dan soal-soal yang ditampilkan secara *online* dari *database* [1].

Pada tahun 2015, Bayu Surya Aditama dkk dalam tesisnya yang berjudul *Aplikasi Ujian Online Berbasis Android* membuat skema yang merujuk pada tesis Alvin, namun dalam penelitian ini digunakan *framework AppsGeyser* untuk membuat aplikasi *Android* berbasis *web* serta *layout* yang dibentuk tidak sesederhana milik Alvin [2].

Pada tahun 2016, Yongki Yonatan Marbun dkk dalam tesisnya yang berjudul *Pembuatan Aplikasi TOEFL Sebagai Media Pelatihan Bahasa Inggris Berbasis Web* membuat sistem informasi tes *TOEFL* berbasis *web* saja. *Web* yang dikembangkan Yongki dkk ini menggunakan Bahasa *PHP* dan *HTML* yang dipadukan dengan *CSS* sebagai *frontendnya*. Sistem informasi ini digunakan sebagai media pelatihan *online* untuk mengukur kemampuan pengguna dalam menghadapi tes *TOEFL* sebenarnya. Sistem informasi ini menggunakan skala *likert* dan pengujian *reliability* pada Algoritma yang digunakan untuk menilai hasil dari tes *TOEFL* yang telah dijawab [3].

Pada tahun 2019 R. Nasser dan P. Saldriani dalam SEMANTIK tahun 2019 yang berjudul Perancangan Dan Pembuatan Aplikasi Simulasi *TOEFL* Berbasis *Desktop* telah membuat aplikasi *TOEFL* yang dijalankan pada PC secara *offline* menggunakan *Adobe Flash Professional CS6* dengan Bahasa pemrograman *Adobe Script 2.0*. Pada Aplikasi ini digunakan metode penilaian menggunakan tabel konversi yang memiliki bobot tertentu dan berbeda-beda terhadap setiap *section test* [4].

Pada tahun 2018 Yolen Perdana Sari dalam Jurnal Informatika Universitas Pamulang yang berjudul Perancangan dan Implementasi Aplikasi *TOEFL* pada Perangkat *Android* berhasil membuat sistem informasi tes *TOEFL* berbasis *Android* yang dikembangkan menggunakan Bahasa *Java* dengan Bahasa *XML* sebagai layoutnya serta *database* yang digunakan berupa *SQLite* yang bersifat *offline*. Yolen menggunakan penilaian menggunakan tabel konversi untuk menentukan bobot soal yang dipakai. Dalam pengembangan sistem ini menggunakan *IDE Eclipse* dengan tambahan *Android SDK*, *Android DT*, dan *Java SE* yang telah terinstall pada *IDE Eclipse* yang digunakan [5].

Pada tahun 2019 Sa'adah dkk dalam jurnal dengan judul Aplikasi Simulasi Tes *TOEFL* Berbasis *Android* di Fakultas Keguruan Dan Ilmu Pendidikan Universitas Hamzanwadi membuat sistem informasi simulasi tes *TOEFL* berbasis *Android* yang isinya berupa soal dan pembahasan serta solusi mengenai tes *TOEFL*. Sistem informasi ini bersifat *offline* sehingga tidak memerlukan koneksi internet [6].

BAB III

LANDASAN TEORI

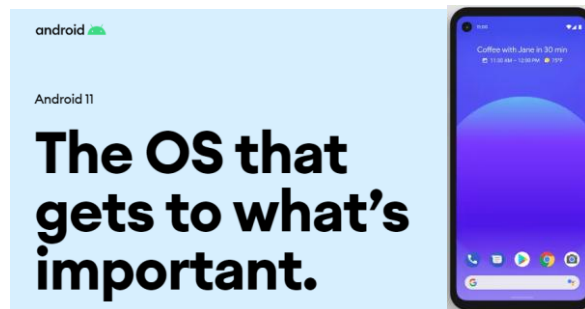
3.1. Sejarah Sistem Operasi *Android*

Android adalah sebuah sistem operasi yang dikembangkan untuk perangkat *mobile* dengan berbasis pada *Linux*. Awalnya, sistem operasi *android* dikembangkan oleh *Android Inc* yang didirikan pada tahun 2003 oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. *Android* diciptakan untuk menyaingi sistem operasi yang populer saat itu seperti *Symbian* dan *Windows Mobile*. Pada saat awal-awal dibentuk, sistem operasi *Android* sempat mengalami jatuh bangun. Sampai pada Agustus 2005, *Google* secara resmi mengakuisisi *Android* dan menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh *Google*. [7]

Dalam pengembangan *android* di tahun 2007, dibentuk sebuah konsorsium bernama *Open Handset Alliance (OHA)* yang terdiri dari beberapa perusahaan yaitu *Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvel Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics Sprint Nextel* dan *T-mobile* yang bertujuan untuk mengembangkan standar terbuka untuk perangkat *mobile*. Setahun kemudian, beberapa perusahaan lain ikut bergabung yaitu *PacketVideo, ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc*. *Android* versi 1.1 yaitu *android* pertama yang diluncurkan oleh *Google* pada tanggal 9 Maret tahun 2009 [1].

Tim *Android* yang dipimpin oleh Andy Rubin kemudian mengembangkan *Android* agar dapat berjalan pada *platform* perangkat seluler berbasis *Kernel Linux*. Melalui *Android*, *Google* berencana untuk masuk ke dalam pasar handphone dunia. Sampai akhirnya pada Oktober 2008, *Google* secara resmi memperkenalkan produk handphone dengan sistem operasi *Android* pertamanya yang bekerjasama dengan *HTC*, yaitu *HTC Dream*. Setelah itu, pada tahun 2010, *Google* resmi merilis *Nexus*

yaitu handphone dengan sistem operasi *Android* yang diproduksi oleh 3 mitranya yaitu *HTC*, *LG*, dan *Samsung* [7].





Gambar 3.1 Sistem Operasi *Android 11* [8]

Gambar 3.1 diatas adalah penampakan dari *System UI* yang dimiliki oleh *Android 11*. Kini, setelah sistem operasi *android* mulai populer, semakin banyak perusahaan merk *handphone* terkenal dunia yang mulai tertarik untuk membuat *handphone* dengan sistem operasi tersebut. Karena bersifat *Open Source*, banyak produsen *handphone* dunia yang bisa dengan bebas mengembangkan kemampuan sistem operasi ini. Pada akhirnya, hingga saat ini sistem operasi *Android* merupakan sistem operasi yang paling berkembang dan memiliki banyak fitur inovatif ketimbang *IOS* [7].

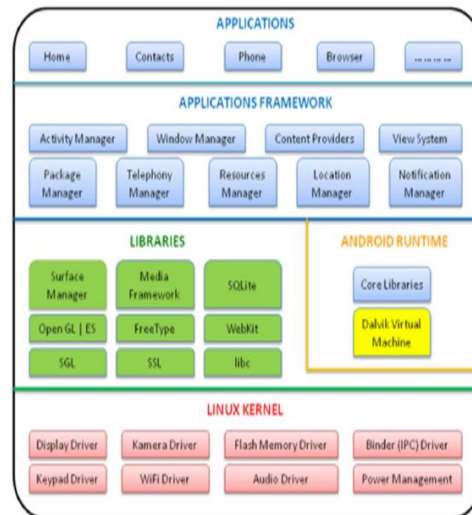
Android dari masa ke masa sudah mengalami beberapa perkembangan yang sangat pesat seperti yang ada pada tabel dibawah ini, kini *android* sudah meluncurkan produk barunya yaitu *Android 11*, serta *Android 12* yang masih berbentuk eksperimental. Tabel 3.1 dibawah ini mencantumkan pembaharuan setiap versi *Android* dari masa ke masa.

Tabel 3.1 Daftar Versi *Android* dari 1.5 – 10 [9]

Version	Code name	Release date	API level	DVM/ART	New features	Icon
10	10	September 3, 2019	29	ART	Live Caption, Smart Reply Sound Amplifier, Dark Theme Privacy & Security, Digital Wellbeing	 android
9	Pie	August 6, 2018	28	ART	Adaptive Battery, Adaptive Brightness	

Version	Code name	Release date	API level	DVM/ART	New features	Icon
8.0-8.1	Oreo	October 25, 2017	26 - 27	ART	Picture-in-Picture	
7.1 - 7.1.2	Nougat	August 22, 2016	24- 25	ART	Multi window, GIF Keyboard	
6.0 - 6.0.1	Marshmallow	October 5, 2015	23	ART	Now On Tap, Permissions Battery (Doze & App Standby)	
5.1 - 5.1.1	Lollipop	November 12, 2014	21	ART	Material Design, Multiscreen, Notifications	
4.4 - 4.4.4	KitKat	October 31, 2013	19 - 20	DVM (and ART 1.6.0)	Voice : Ok Google, Immersive Design, Smart Dialer	
4.1 - 4.3.1	Jelly Bean	July 9, 2012	18	DVM	Google Now, Actionable Notifications, Account Switching	
4.0 - 4.6	Ice Cream Sandwich	October 19, 2011	15	DVM	Custom Home Screen, Data Usage Control, Android Beam	
3.0 - 3.2.6	HoneyComb	February 22, 2011	11 - 13	DVM	Tablet-Friendly Design System Bar, Quick Settings	
2.3 - 2.3.7	Gingerbread	February 9, 2011	9	DVM	Gaming APIs, NFC Battery Management	
2.2 - 2.2.3	Froyo	May 20, 2010	8	DVM	Voice Action Portable Hotspot Dalvik JIT	
2.0 - 2.1	Eclair	October 26, 2009	5	-	Google Maps Navigation Home Screen Customization Speech-to-Text	
1.6	Donut	September 15, 2009	4	-	Quick Search Box Screen Size Diversity Android Market	
1.5	Cupcake	April 27, 2009	3	-	-	

Secara umum arsitektur utama pada *android* itu sama di setiap versinya, sehingga arsitektur *android* dapat digambarkan seperti pada Gambar 3.2 dibawah.

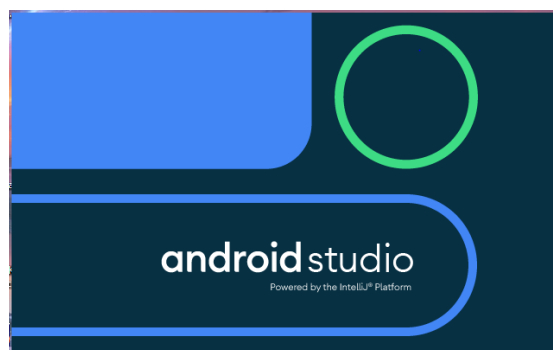


Gambar 3.2 Arsitektur *Android* (Murphi, *Beginning Android*, 2009) [5]

3.2. Pemrograman Aplikasi *Android*

3.2.1. *Android Studio*

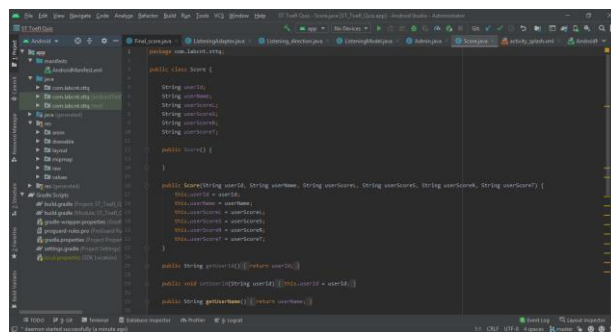
Android Studio adalah *Integrated Development Environment (IDE)* resmi untuk pengembangan aplikasi *Android*, yang didasarkan pada *IntelliJ IDEA* seperti pada Gambar 3.3.



Gambar 3.3 Logo *Startup Android Studio IDE*

Selain sebagai editor kode dan fitur *developer IntelliJ* yang andal, *Android Studio* menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi *Android*, seperti:

- a. Sistem *build* berbasis *Gradle* yang fleksibel
- b. *Emulator* yang cepat dan kaya fitur
- c. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat *Android*
- d. Terapkan Perubahan untuk melakukan *push* pada perubahan kode dan *resource* ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
- e. *Template kode* dan integrasi *Git* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
- f. *Framework* dan alat pengujian yang lengkap
- g. Alat lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
- h. Dukungan *C++* dan *NDK*
- i. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine* [10].



Gambar 3.4 GUI *Android Studio*

Penampakan dari *GUI Android Studio* sendiri mirip dengan IDE besutan JetBrains lainnya seperti yang ditunjukkan oleh Gambar 3.4. Setiap project di *Android Studio* berisi satu atau beberapa modul dengan *file* kode sumber dan *file resource*. Jenis modul meliputi:

- a. Modul aplikasi *Android*
- b. Modul *library*
- c. Modul *Google App Engine*
- d. Semua file *build* terlihat di tingkat teratas di bagian *Gradle Script* dan setiap modul aplikasi berisi folder berikut:
- e. *manifest*: Berisi file *AndroidManifest.xml*.
- f. *java*: Berisi file kode sumber *Java*, termasuk kode pengujian *JUnit*.
- g. *res*: Berisi semua *resource* non-kode, seperti tata letak *XML*, *string UI*, dan gambar *bitmap* [10].

3.2.2. *Java*

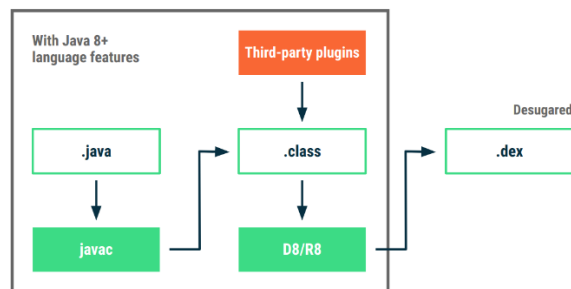
Java adalah sebuah bahasa yang diciptakan oleh James Gosling di tahun 1990-an dengan logo berbentuk cangkir kopi seperti pada Gambar 3.5. *Java* muncul sebagai bahasa yang dapat dijalankan di berbagai *platform* tanpa perlu melakukan re-kompilasi. Berdasarkan *TIOBE Programming Community Index* yang meninjau popularitas bahasa pemrograman, *Java* masih menjadi bahasa pemrograman nomor satu di dunia. Data dari *Oracle* menyatakan bahwa bahasa *Java* digunakan 90% perusahaan terkemuka yang masuk dalam daftar *Fortune 500*. Bahasa *Java* dapat digunakan untuk mengembangkan aplikasi pada *platform desktop*, *web*, *mobile*, hingga *embedded* dan *IoT* [11].



Gambar 3.5 Logo Bahasa *Java*

Plugin Android Gradle 3.0.0 mendukung semua fitur bahasa *Java 7* dan sebagian fitur bahasa *Java 8* yang berbeda menurut versi *platform*. Saat membuat aplikasi menggunakan *plugin Android Gradle 4.0.0* dan yang lebih tinggi dapat

menggunakan sejumlah *API* bahasa *Java 8* tanpa memerlukan *API level minimum* untuk aplikasi Anda. *Plugin Android Gradle* menyediakan dukungan bawaan untuk menggunakan fitur bahasa *Java 8* tertentu dan *library* pihak ketiga yang menggunakannya. *Toolchain default* yang digambarkan pada Gambar 3.6 mengimplementasikan fitur bahasa baru dengan menjalankan transformasi *bytecode*, yang disebut *desugar*, sebagai bagian dari kompilasi *file class D8/R8* menjadi kode *dex* [12].



Gambar 3.6 *Toolchain Java* [12]

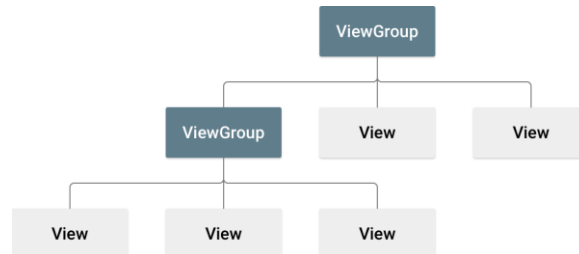
3.2.3. XML

Extensible Markup Language (XML) adalah sebuah *markup language* yang menyediakan deskripsi, penyimpanan, dan format transmisi untuk pertukaran data melalui *webservices*. *XML* mirip dengan *HyperText Markup Language (HTML)* dimana keduanya memiliki elemen \pm elemen, *attribute* \pm *attribute* dan nilai \pm nilai. Perbedaan *XML* dan *HTML* yaitu *XML* didesain untuk mengirim dan menyimpan data dan berfokus kepada data apa yang dikirim atau disimpan, sedangkan *HTML* didesain untuk menampilkan data dan berfokus kepada bagaimana data tersebut ditampilkan. Singkatnya, *XML* adalah tentang membawa informasi sedangkan *HTML* adalah tentang menampilkan informasi [1].

3.2.4. Layout

Layout digunakan untuk menentukan struktur dari antarmuka pengguna di aplikasi, seperti di dalam aktivitas. Semua elemen pada tata letak dibuat menggunakan hierarki objek *View* dan *ViewGroup*. *View* biasanya menggambarkan sesuatu yang terlihat, dan pengguna dapat berinteraksi dengannya. Sedangkan

ViewGroup adalah *container* tak terlihat yang menentukan struktur tata letak bagi *View* dan objek *ViewGroup* lainnya, seperti yang ditampilkan pada Gambar 3.7 [13].



Gambar 3.7 Hierarki *Layout* Dalam Mendefinisikan *UI* [13]

Terdapat 2 cara dalam mendeklarasikan layout yaitu :

- a. Deklarasikan elemen *UI* dalam *XML*. *Android* menyediakan kosakata *XML* sederhana yang sesuai dengan class dan subclass layout, seperti halnya untuk *widget* dan layout
- b. Buat *instance* elemen tata letak pada saat *runtime*. Aplikasi Anda dapat membuat objek Tampilan dan *ViewGroup* (serta memanipulasi propertinya) secara terprogram

Framework Android memiliki fleksibilitas untuk menggunakan salah satu atau kedua metode ini untuk membuat *UI* aplikasi. Misalnya, kita dapat mendeklarasikan tata letak default aplikasi pada *XML*, kemudian memodifikasinya saat *runtime* [13].

3.2.4.1. *ViewGroup*

Seperti pada sub bab 3.2.4 bahwa *ViewGroup* adalah *container* yang berisi *View*. *ViewGroup* tidak terlihat namun, setiap *ViewGroup* berbeda satu sama lainnya. Sifat dari *layout* sama seperti kertas, maka *View Group* diibaratkan seperti wadah untuk meletakkan *View* dalam pengaturan tertentu.

- a. *Linear Layout*

Dengan menggunakan *Linear Layout* setiap komponen atau elemen yang akan digunakan untuk merancang aplikasi *android* disejajarkan dalam satu arah saja, yaitu secara *horizontal* atau *vertical* [14].

b. *Relative Layout*

Relative Layout merupakan layout yang bisa dipakai untuk mengatur widget atau komponen aplikasi *android* secara *relative* (bebas), tidak sebatas *vertical* atau *horizontal* saja [14].

c. *Table Layout*

Untuk merancang *layout* menggunakan baris dan kolom kita bisa menggunakan *Table Layout*. Tidak akan ada garis kolom, baris, atau *cell* yang ditampilkan meskipun namanya adalah *table layout* [14].

d. *Frame Layout*

Jika kita ingin membuat *layout* dengan komponen-komponen yang saling tumpang tindih, maka kita bisa menggunakan *Frame Layout*. Misal kita ingin memasukan komponen tombol ke dalam komponen gambar [14].

e. *Constraint Layout*

Constraint Layout adalah *layout* yang baru di *android studio*. Ia merupakan pengembangan dari *Linear Layout*. Dengan menggunakan *Constraint Layout* kita bisa membuat tampilan aplikasi *android* yang *responsive*, *kompleks*, dan *powerfull* [14].

3.2.4.2. View

View adalah deklarasi untuk mendefinisikan suatu objek seperti gambar, *bitmap*, tombol, dan lain lain.

a. *ImageView*

ImageView sesuai namanya berfungsi untuk menampilkan gambar yang tersimpan didalam folder *drawable*. Gambar akan muncul dan terdefinisi didalam *ImageView* menggunakan perintah *android:src="@drawable/nama_file"*.

b. *TextView*

TextView berfungsi untuk menampilkan *string* pada aktivitas. *String* biasanya tersimpan didalam file *string.xml* pada folder *value*.

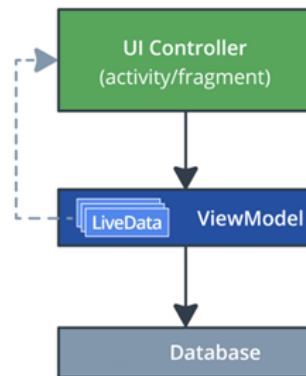
Sehingga untuk mendefinisikan *TextView* menggunakan perintah `android:text="string yang ingin ditulis"` apabila ditulis langsung didalam *view* atau `android:text="@string/variable_string"` apabila terdapat pada file *string.xml*.

c. *RecyclerView*

RecyclerView adalah tampilan yang menggunakan arsitektur yang disederhanakan dengan *UI controller*, *ViewModel*, dan *LiveData*. Manfaat terbesar dari *RecyclerView* adalah sangat efisien untuk daftar besar. Secara default, *RecyclerView* hanya berfungsi untuk memproses atau menggambar item yang saat ini terlihat di layar. Misalnya, jika list memiliki seribu elemen tetapi hanya 10 elemen yang terlihat, *RecyclerView* hanya berfungsi untuk menggambar 10 item di layar.

Ketika pengguna melakukan *scroll*, *RecyclerView* mengetahui item baru apa yang seharusnya ada di layar dan tidak cukup berfungsi untuk menampilkan item itu. Ketika suatu *item* di *scroll* dari layar, tampilan item tersebut didaur ulang. Itu berarti item diisi dengan konten baru yang *scroll* ke layar. Perilaku *RecyclerView* ini menghemat banyak waktu pemrosesan dan membantu *scroll list* dengan lancar. Ketika suatu item berubah, alih-alih menggambar ulang seluruh daftar, *RecyclerView* dapat memperbarui satu item itu. Ini adalah keuntungan efisiensi yang sangat besar ketika menampilkan daftar item kompleks [15].

Dalam urutan yang ditunjukkan pada Gambar 3.8, kita dapat melihat bahwa satu tampilan telah diisi dengan data, ABC. Setelah itu tampilan bergulir dari layar, *RecyclerView* menggunakan kembali tampilan untuk data baru, XYZ.



Gambar 3.8 Mekanisme *RecyclerView* Bekerja

d. *Button*

Button terdiri dari teks atau ikon (atau teks dan ikon) yang menyampaikan tindakan yang terjadi saat pengguna menyentuhnya. Saat kita ingin memberikan suatu perintah/tindakan pada *Button*, kita perlu *menulis method* `onClick` apabila kita menggunakan fitur `setOnClickListener` pada *class* terkait. Kita juga dapat mendefinisikan terlebih dahulu apa yang akan dilakukan pada *Button* menggunakan `android:onClick` [16].

e. *RadioButton* (Tombol Pilihan)

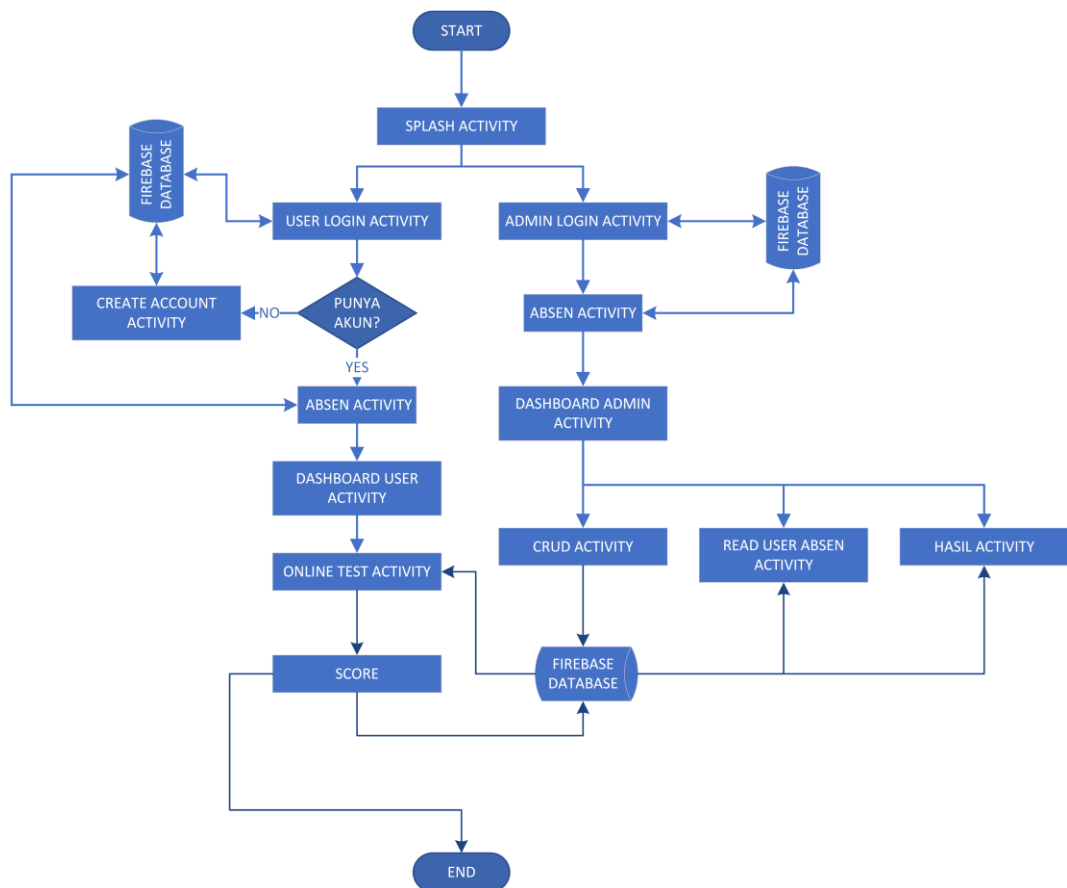
Tombol pilihan memungkinkan pengguna memilih salah satu dari beberapa opsi. Kita sebaiknya menggunakan tombol pilihan untuk beberapa opsional yang eksklusif satu sama lain jika *menurut* kita pengguna perlu melihat semua opsi yang tersedia secara berdampingan. Jika tidak perlu menampilkan semua opsi secara berdampingan, sebagai gantinya kita bisa menggunakan *spinner*. Untuk membuat setiap opsi tombol pilihan, kita dapat membuat *RadioButton* dalam *layout*. Namun, karena tombol pilihan bersifat eksklusif satu sama lain, kita harus mengelompokkannya didalam *RadioGroup*. Dengan mengelompokkannya, sistem memastikan bahwa hanya satu tombol pilihan yang dapat dipilih dalam satu waktu [17].

BAB IV

PEMBAHASAN

4.1. Alur Kegiatan

Dalam melakukan kegiatan membuat aplikasi bergerak berbasis *android* tentu perlu dibuat sebuah rancangan terlebih dahulu, supaya aplikasi yang dibuat memiliki target yang jelas untuk dicapai. Begitu juga dengan aplikasi UNOT ini yang memiliki rancangan seperti Gambar 4.1.



Gambar 4.1 Flowchart UNOT

Seperti yang telah dijelaskan sebelumnya aplikasi bergerak yang akan kami buat memiliki rancangan seperti pada Gambar 4.1. Secara garis besar aplikasi ini mempunyai dua bagian utama, yaitu bagian *admin* dan bagian *user*. Setiap bagian memiliki peranannya sendiri dalam *menunjang* jalanya aplikasi ini.

4.1.1. *Admin Activity*

Admin Activity merupakan bagian yang dibuat untuk kebutuhan *admin*, seperti pembuatan soal, melihat nilai yang didapatkan oleh *user* dan lainnya. Didalam *Admin Activity* ini memiliki bagian-bagiannya sendiri, yaitu :

a. *Admin Login Activity*

Activity ini merupakan halaman awal dari bagian *admin*, berfungsi untuk melakukan pengecekan *username* dan *password* akun *admin*, jika data yang dimasukan benar maka akan berpindah ke *Dashboard Admin Activity*, dan jika data yang dimasukan salah maka halaman tidak akan berpindah ke *Dashboar Admin Activity*.

b. *Dashboard Admin Activity*

Activity ini berfungsi untuk menyimpan dan menampilkan *menu-menu* yang dapat digunakan oleh *admin*, seperti *CRUD Activity*, *Read User Absen Activity*, dan *Hasil Activity*.

c. *CRUD Activity*

Activity ini merupakan halaman yang berfungsi sebagai editor untuk soal yang akan ditampilkan ke halaman *user*. Di halaman ini dapat dilakukan pembuatan soal, *update* soal, dan menampilkan *preview* dari soal yang telah dibuat.

d. *Read User Absen Activity*

Activity ini bertugas untuk menampilkan daftar hadir atau absen dari *user* yang mengikuti *test* melalui aplikasi UNOT.

e. *Hasil Activity*

Dan yang terakhir dari bagian *admin* adalah *Hasil Activity*, *Activity* ini bertanggung jawab untuk menampilkan hasil atau nilai yang didapatkan oleh *user test*.

4.1.2. *User Activity*

User Activity merupakan *activity* yang bertanggung jawab terhadap *user* atau peserta *test*. *User Activity* ini dibagi menjadi beberapa bagian, yaitu :

a. *User Login Activity*

Activity ini merupakan halaman yang bertanggung jawab untuk memeriksa *username* dan *password* dari akun peserta. Jika peserta telah memiliki akun maka hanya perlu memasukan *username* dan *password* yang telah terdaftar. Apabila peserta belum memiliki akun yang terdaftar maka dapat mendaftar terlebih dahulu. Jika sudah selesai mendaftar maka Langkah selanjutnya adalah melakukan *login*.

b. *Create Account Activity*

Activity ini bertugas untuk melakukan pendaftaran akun bagi peserta yang belum memiliki akun untuk masuk kedalam aplikasi UNOT.

c. *Absen Activity*

Bagian ini merupakan bagian absen dari peserta *test*. Peserta yang ingin melakukan *test* dengan menggunakan aplikasi UNOT perlu melakukan absen terlebih dahulu, supaya dapat dilakukan pengecekan oleh *admin*, siapa saja yang telah *login* kedalam aplikasi.

d. *Dashboard User Activity*

Sama seperti pada bagian *admin*, *activity* ini berfungsi untuk menampilkan *menu-menu* yang dapat diakses oleh peserta *test*, seperti info akun, dan bahkan bisa membuka siacad dengan menggunakan *menu webview* yang terdapat pada *dashboard* peserta.

e. *Online Test Activity*

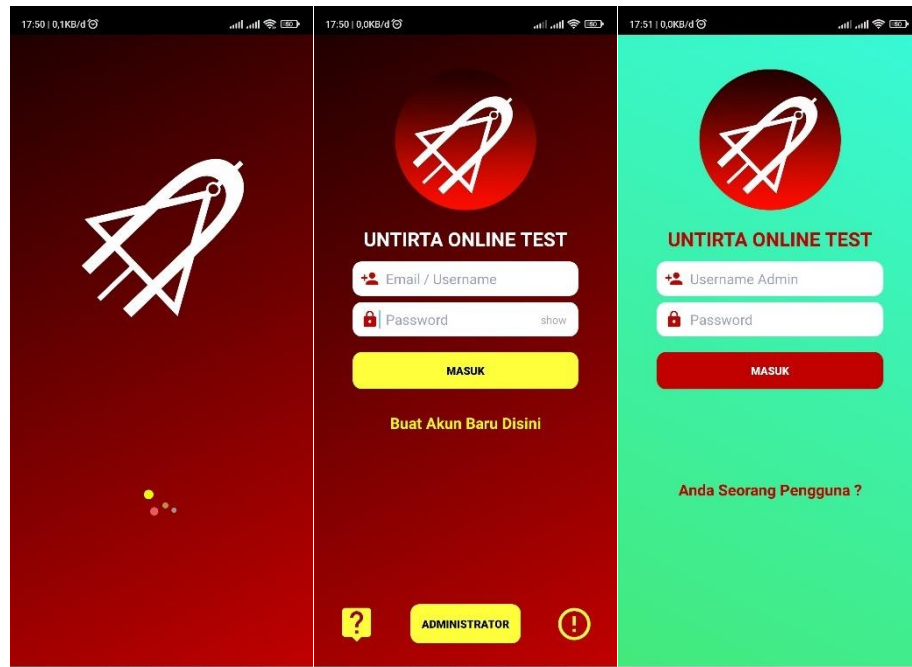
Activity ini merupakan *activity* utama dari aplikasi ini, yaitu *activity* yang bertugas untuk menampilkan soal yang akan dikerjakan oleh peserta.

f. *Score Activity*

Dan yang terakhir dari bagian *user* adalah *score activity*, *activity* ini berfungsi sebagai tempat untuk menampilkan hasil atau nilai yang didapatkan peserta setelah mengerjakan soal yang ada pada *Online Test Activity*.

4.2. Struktur Aplikasi

Dalam pembuatan aplikasi ini kami telah berhasil membuat layout beserta fungsinya sesuai dengan rancangan yang ada pada Gambar 4.1. Layout aplikasi yang pertama telah kami buat adalah *Splash*, *Login User*, dan *Login Admin* seperti pada Gambar 4.2.



Gambar 4.2 *Splash*, *Login User*, dan *Login Admin*

Seperti yang terlihat pada Gambar 4.2, *activity* yang telah dibuat adalah splash screen, *login user*, dan *login admin*. Splash screen merupakan tampilan awal ketika membuka aplikasi UNOT. Kemudian *login user* adalah *activity* untuk melakukan *login* ke aplikasi dengan menggunakan *username* yang berupa email untirta dan *password*, *activity login user* ini terhubung dengan *firebase authentication* untuk melakukan pengecekan akun yang akan digunakan untuk *login* kedalam halaman *user*. Jika akun yang digunakan *user* terdapat pada *firebase authentication* maka proses *login* berhasil. Jika pada *firebase authentication* tidak terdapat akun yang sama dengan yang dimasukan oleh *user* maka *user* tersebut tidak bisa *login* kedalam aplikasi.

Dalam *activity login user* terdapat button *administrator*, fungsi dari button tersebut adalah untuk mengarahkan pengguna aplikasi ke *activity login admin*. Berbeda dengan *activity login user*, pada *activity login admin* ini tidak menggunakan *username* berupa email untirta maupun terhubung dengan *firebase authentication*, akan tetapi menggunakan *firebase realtime database* sebagai tempat penyimpanan akun *admin*, proses yang terjadi pada *activity* ini hampir sama seperti pada *activity login user*, pertama pengguna akan memasukan *username* dan *password* akun *admin*, jika data akun yang dimasukan sama dengan data akun yang terdapat pada *firebase realtime database* maka proses *login admin* berhasil, akan tetapi jika didalam *firebase realtime database* tidak ditemukan data akun yang cocok dengan data akun yang dimasukan pengguna maka proses *login* akan gagal. Setelah proses *login admin* selesai maka akan diarahkan ke *dashboard admin* seperti Gambar 4.3.

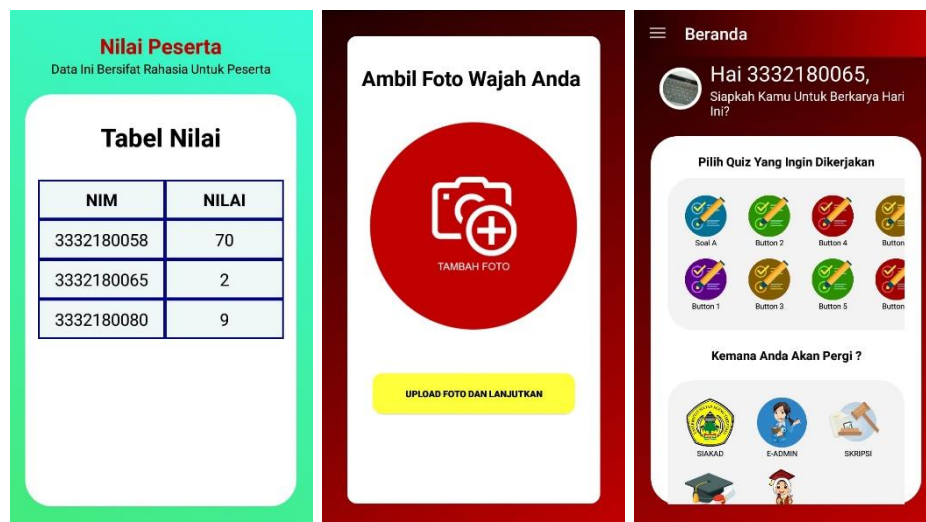


Gambar 4.3 *Dashboard Admin, CRUD Soal, Kontrol Aplikasi*

Seperti yang telah dijelaskan sebelumnya, *dashboard admin* merupakan tempat untuk menampilkan *menu-menu* yang dapat digunakan oleh *admin*, layout untuk *dashboard admin* dapat dilihat pada Gambar 4.3. Didalam *dashboard admin* terdapat beberapa *menu*, yaitu *control* aplikasi, *CRUD* soal, dan lihat nilai. Kontrol aplikasi berfungsi untuk membuka dan menutup aplikasi, jika aplikasi sedang tidak diperlukan untuk kebutuhan *test* maka dapat dipilih matikan layanan aplikasi, maka

user tidak akan bisa mengakses online *test* qctivity, dan jika aplikasi akan digunakan untuk *test* online maka dapat dipilih Hidupkan Layanan Quiz maka peserta dapat mengakses onile *test* activity.

Selain *control* aplikasi, didalam *dashboard admin* juga terdapat *CRUD* soal. *Activity* ini merupakan *activity* yang digunakan untuk memasukan soal, dan mengupdate soal yang diinginkan. Setelah itu dapat dilakukan preview untuk melihat soal yang dibuat apakah sudah sesuai denga napa yang diinginkan atau belum. Kemudian didalam *dashboard admin* juga terdapat *menu* untuk meilhat nilai yang didapatkan oleh peserta, seperti yang terlihat pada Gambar 4.4.

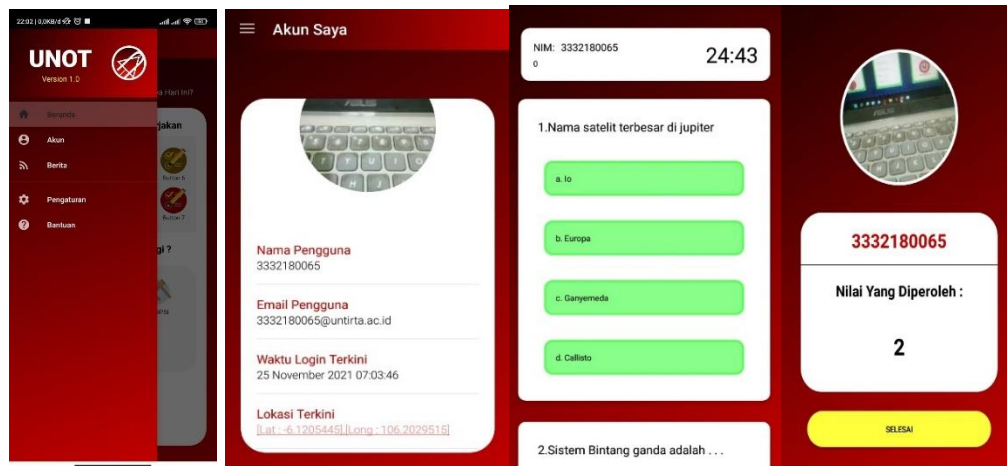


Gambar 4.4 Nilai Peserta, Absen Peserta, dan *Dashboard* Peserta

Menu terakhir yang terdapat pada *dashboard admin* adalah *menu* untuk meilhat nilai peserta yang mengikuti *test* seperti pada Gambar 4.4. didalam *activity* nilai peserta terdapat table yang menampilkan NIM dan hasil yang didapatkan oleh peserta tersebut.

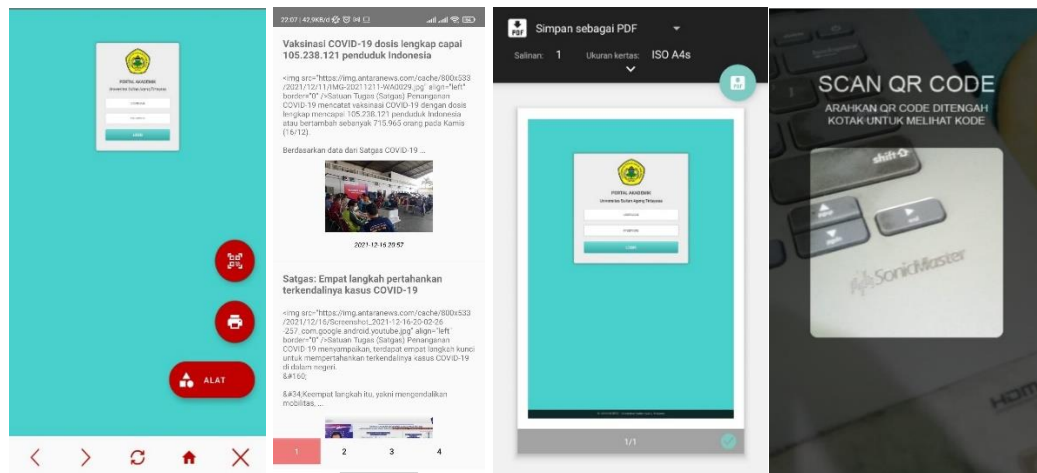
Activity selanjutnya adalah *activity* absen peserta, pada *activity* ini peserta diharuskan melakukan mengambil foto, jika tidak maka tidak akan bisa lanjut ke halaman selanjutnya, gambar yang diambil daari *activity* ini akan disimpan kedalam *firebase storage* dan pathnya disimpan dalam *firebase realtime database*. Setelah melakukan absen maka akan berpindah ke *dashboard* peserta. Pada *dashboard* ini terdapat banyak *menu* seperti yang terlihat pada Gambar 4.4. dari *dashboard* peserta

ini terdapat symbol 3 garis pada bagian kiri atas, symbol tersebut berfungsi untuk menampilkan info pribadi peserta seperti pada gambar 4.5.



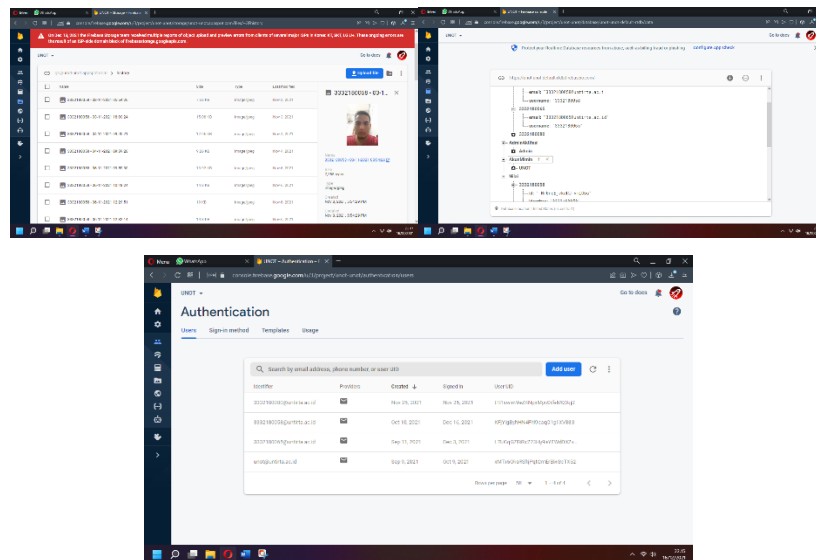
Gambar 4.5 Info Akun, *Test Activity*, dan Hasil *Activity*

Jika kita klik ikon pojok kiri atas maka drawer akan muncul dan menampilkan beberapa menu yang salah satunya terdapat informasi akun seperti pada Gambar 4.5. Informasi akun ini berisi informasi pribadi dari peserta, seperti foto hasil dari absen, kemudian NIM peserta, email peserta, waktu *login* peserta, dan lokasi terkini peserta. Kemudian pada *dashboard* peserta terdapat *menu* untuk melakukan *test* online, ketika *menu* tersebut ditekan maka akan masuk ke bagian soal. Pada bagian soal ini terhubung dengan *firebase realtime database*, sehingga soal yang ditampilkan merupakan soal yang disimpan oleh *admin* kedalam *firebase realtime database*. Lalu terdapat juga informasi seperti NIM dan waktu yang tersisa untuk mengerjakan soal. Setelah menyelesaikan semua soal maka akan dipindahkan ke hasil *activity*. Pada *activity* ini akan ditampilkan hasil yang diperoleh oleh peserta beserta NIMnya, kemudian hasil dan NIM tersebut akan disimpan kedalam *firebase realtime database* sehingga dapat terlihat oleh *admin*. Selain *menu-menu* yang telah disebutkan tadi, terdapat juga *menu* yang menggunakan *webview* seperti Gambar 4.6.



Gambar 4.6 *Webview Siakad, Feed, Save pdf, Scan QR Code*

Pada aplikasi bergerak berbasis *android* kita dapat menggunakan *webview* untuk masuk ke web yang telah ditemukan menggunakan aplikasi yang kita buat, seperti yang terlihat pada Gambar 4.6, kami menggunakan *webview* untuk membuka situs akademik yang ada di UNTIRTA seperti Siakad, Spada, E-Administrasi, Wisuda, dll. Pada tampilan *webview* tersebut terdapat beberapa *tool* yang disematkan pada *Floating Button* yaitu *Save pdf*, dan *Scan QR Code*. *Save pdf* berfungsi untuk menyimpan tampilan halaman web yang saat ini dibuka kedalam file berformat PDF, sedangkan *Scan QR Code* dapat digunakan untuk melakukan *Scan QR* untuk melakukan absen di siakad, ketika kode absen berhasil discan maka bisa dilakukan copy kemudian dipaste kedalam presensi siakad. Selain itu, pada menu drawer terdapat *webview* lainnya saat kita klik drawer berita kita akan diarahkan pada halaman feed berita nasional diindonesia. Feed berita ini ditambahkan selain memperkaya fitur aplikasi juga dapat menambah wawasan user terkait informasi di Indonesia.



Gambar 4.7 Firebase Database

Sebelumnya telah dibahas sedikit mengenai Firebase Database dan penggunaannya diparagraf sebelumnya kami selanjutnya akan membahas hasil data yang tersimpan di Firebase seperti pada Gambar 4.7. Setelah data tersimpan di firebase, semua data akan tersimpan dalam format JSON di server. Mirip dengan SQL, file JSON juga menggunakan query untuk CRUD dan struktur yang sama, hanya saja JSON jauh lebih sederhana dalam melakukan CRUD karena primary key yang digunakan jauh lebih fleksibel dan tidak selalu terikat seperti SQL. Perbedaan lainnya adalah penyimpanan tipe data seperti Blob disimpan pada Firebase Storage dan id dari file yang disimpan di Storage akan tertulis di Firebase Database.

Dalam melakukan CRUD pada Firebase Storage, kita dibebaskan untuk menggunakan berbagai cara baik menggunakan id maupun identifikasi file secara langsung berdasarkan path file yang ada di Firebase Storage. Kebebasan lainnya juga ada pada fitur Firebase Authentication yang mana kita tidak perlu membuat table data lagi dan cukup untuk mengaktifkan fitur ini dan didefinisikan di kode program aplikais milik kita maka data akan langsung tersimpan. Dalam hal ini pun kita dapat menyimpan data user di authentication maupun di database tergantung bagaimana cara kalian mengolah data pada program yang sedang ditulis.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil yang telah dicapai dari project mata kuliah ini sesuai dengan tujuan utama dari tema yang kami ambil yaitu Pembangunan Sistem Informasi UNOT berbasis *Android*, sehingga kesimpulan yang dapat kami ambil yaitu :

- a. Aplikasi “*UNOT*” yang merupakan sistem informasi tes *online* ini belum bersifat *final* karena masih dapat dikembangkan lagi ketahap berikutnya dengan fitur yang jauh lebih kompleks lagi.
- b. Hubungan antara Bahasa *Java* dan *XML* yaitu pada *OOP (Object Oriented Program)* yang mana *XML* sebagai *layout* yang berisi berbagai macam *object* dan tata letak yang telah diatur, sehingga Bahasa *Java* dapat melakukan orientasi objek secara maksimal.
- c. Aktivitas (*Activity*) merupakan kesatuan dari *layout* dan *main program*, sehingga dalam pembangunan aplikasi *android* meskipun berfokus pada *layout (xml)* sebagai *front-end* dan *firebase database* sebagai *back-end*, akan tetapi, *main program* merupakan keseluruhan dari *front-end* dan *back-end* agar aplikasi dapat berfungsi dengan semestinya dalam menampilkan objek.
- d. Penggunaan *IDE Android Studio* sangat efisien ketimbang menggunakan *IDE* lain untuk mengembangkan aplikasi *android* karena bersifat *real time* dalam menampilkan letak objek pada *layout* serta proses *compiling* yang sangat mudah, sehingga proses pengembangan aplikasi dapat selesai lebih cepat.
- e. Penggunaan *Firebase Database* jauh lebih mudah dan lebih *user-friendly* ketimbang menggunakan *database* lain seperti *SQL* atau *SQLite* karena *Firebase* selain gratis juga memiliki kompatibilitas penuh pada platform *Android*. Berbeda dengan *SQLite* dan *SQL* yang membutuhkan server hosting agar *database* bekerja secara *realtime* dan online.

5.2. Saran dan Kritik

Kami memberikan beberapa saran dan kritik yang membangun pada diri kami sendiri agar kedepannya aplikasi “*UNOT*” dapat berkembang lagi. Berikut ini adalah beberapa kritik dan saran yang telah didapatkan :

- a. *Design* untuk *show* dan *hide password* pada *Login Activity* dan *Create Account Activity* dapat diganti menggunakan simbol berupa gambar mata.
- b. Pilihan ganda saat diklik langsung menghilang tanpa ada notifikasi serta tidak bisa diubah kembali. Diusahakan kedepannya Pilihan ganda saat dipilih dapat diubah sesuka hati atau setidaknya terdapat dialog untuk membatalkan jawaban sebelumnya.
- c. Aplikasi dapat dikembangkan lebih baik lagi dengan memasukan banyak Mata Kuliah dan beberapa akun *Administrator* dan lingkungan baru agar setiap *Admin* tidak mengalami bentrok dalam melakukan *CRUD* pada aplikasi.

DAFTAR PUSTAKA

- [1] A. A. Leiman, A. Handojo and A. Noertjahyana, "APLIKASI UJIAN ONLINE PADA MOBILE DEVICE *ANDROID*," *Jurnal Tesis Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra*, 2013.
- [2] B. S. Aditama and H. , "APLIKASI UJIAN ONLINE BERBASIS *ANDROID*," *Undergraduate Theses from JBPTUNIKOMPP*, 2015.
- [3] Y. Y. Marbun, R. R. Isnanto and K. T. Martono, "PEMBUATAN APLIKASI TOEFL SEBAGAI MEDIA PELATIHAN BAHASA," *Jurnal Teknologi dan Sistem Komputer UNDIP*, vol. 4, no. 1, pp. 83-92, 2016.
- [4] R. Nasser and P. Saldriani, "Perancangan Dan Pembuatan Aplikasi Simulasi TOEFL," *SEMANTIK*, pp. 184-191, 2019.
- [5] Y. P. Sari, "Perancangan dan Implementasi Aplikasi TOEFL pada Perangkat," *Jurnal Informatika Universitas Pamulang*, vol. 3, no. 3, pp. 1-7, 2018.
- [6] B. D. Dwi Arianti and Sa'adah, "Aplikasi Simulasi Tes Toefl Berbasis *Android* di Fakultas Keguruan Dan Ilmu Pendidikan Universitas Hamzanwadi," *EDUMATIC: Jurnal Pendidikan Informatika*, vol. 3, no. 1, pp. 21-27, 2019.
- [7] Microdata Indonesia Corporation, "Sejarah *Android*," microdataindonesia, 29 Juli 2019. [Online]. Available: <http://microdataindonesia.co.id/news/read/202/sejarah-android>. [Accessed 15 Desember 2021].
- [8] *Android* Inc, "What is *Android*?," Google, [Online]. Available: https://www.android.com/intl/id_id/what-is-android/. [Accessed 15 Desember 2021].
- [9] Dicoding Academies Module, "Belajar Membuat Aplikasi *Android* untuk Pemula," dicoding, 06 September 2017. [Online]. Available: www.dicoding.com/academies/51. [Accessed 15 Desember 2021 [Saved Web at : 21/12/2019]].
- [10] *Android* Inc with Developer Team, "Mengenal *Android* Studio," Google Inc, [Online]. Available:

<https://developer.android.com/studio/intro?hl=id>.

[Accessed 15 Desember 2021].

- [11] Dicoding Academies Indonesia, "Memulai Pemrograman Dengan Java," Dicoding, [Online]. Available: <https://www.dicoding.com/academies/60>. [Accessed 15 Desember 2021].
- [12] *Android* Inc with Developer Team, "Menggunakan fitur dan API bahasa Java 8," Google, [Online]. Available: <https://developer.android.com/studio/write/java8-support?hl=id>. [Accessed 15 Desember 2021].
- [13] *Android* Inc with Developer Team, "Tata Letak," Google, [Online]. Available: <https://developer.android.com/guide/topics/ui/declaring-layout>. [Accessed 15 Desember 2021].
- [14] Guntoro, "Belajar Menggunakan 5 Jenis Layout di *Android* Studio," Badoy Studio, 19 Januari 2021. [Online]. Available: <https://badoystudio.com/layout-android-studio/>. [Accessed 16 Desember 2021].
- [15] Sudarmanto, "Praktikum Pemrograman Berbasis Mobile, prodi TI," 2019. [Online]. Available: <https://darmanto.akakom.ac.id/praktikandroid/Modul%205%20Praktikum%20Pemrograman%20Mobile.pdf>. [Accessed 16 Desember 2021].
- [16] *Android* Inc with Developers Team, "Tombol," Google, *Android* Inc, [Online]. Available: <https://developer.android.com/guide/topics/ui/controls/button?hl=id>. [Accessed 16 Desember 2021].
- [17] *Android* Inc With Developers Team, "Tombol Pilihan," Google, *Android* Inc, [Online]. Available: <https://developer.android.com/guide/topics/ui/controls/radiobutton?hl=id>. [Accessed 16 Desember 2021].

LAMPIRAN

SOURCE CODE PROGRAM UNTIRTA ONLINE TEST (UNOT)



<https://github.com/kuro-kazuto/UNOT>

Github Kami :

- **Galih** : <https://github.com/kuro-kazuto>
- **Dimas** : <https://github.com/Dimas014>