

# Rancang dan Bangun Aplikasi Android Mobile Edunex

Reza Hadi Fairuztama  
Sekolah Teknik Elektro & Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
rreza510@gmail.com

Prof. Yusep Rosmansyah, S.T., M.Sc.,  
Ph.D.  
Sekolah Teknik Elektro & Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
yusep@stei.itb.ac.id

Dicky Prima Satya, S.T., M.T.  
Sekolah Teknik Elektro & Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
dicky@informatika.org

**Abstrak**—Pada awal tahun 2020, seluruh dunia mengalami pandemi Covid-19 yang mengakibatkan berbagai kegiatan bersama dilakukan secara jarak jauh atau daring. Hal ini mendorong teknologi sebagai salah satu aspek yang sangat penting dalam berkomunikasi, utamanya teknologi informasi. Salah satu kegiatan yang terimbas adalah kegiatan ajar mengajar. Untuk persiapan pembelajaran jarak jauh, Institut Teknologi Bandung (ITB) telah membuat dan mengembangkan suatu sistem internal yang merupakan sebuah Learning Management System (LMS), bernama Edunex. Untuk versi selanjutnya sudah ada rencana untuk membuat versi mobile app. Hal ini diharapkan membantu membuat sistem Edunex lebih fleksibel dan menyesuaikan preferensi pengguna, dan mendukung sistem pembelajaran bauran (hybrid) yang ingin diadakan ITB dalam waktu yang mendatang. Untuk menunjang pengembangan ini, dilakukan project capstone secara kelompok untuk mengembangkan mobile app tersebut, dengan dokumen ini menjadi laporan pengembangan aplikasi versi Android. Perancangan aplikasi dilakukan dengan melakukan pendefinisian fitur-fitur yang akan dikembangkan, pendefinisian kebutuhan fungsional dan non fungsional, merancang diagram use case, dan merancang diagram aktivitas. Pembangunan dilakukan menggunakan metode Agile scrum dengan melakukan 11 sprint, dengan membuat aplikasi Android native dengan arsitektur UI-Domain-Data. Aplikasi menggunakan server backend yang sama dengan aplikasi situs web.

**Keywords**—*component, formatting, style, styling, insert (key words)*

## I. PENDAHULUAN

Pada awal tahun 2020, seluruh dunia mengalami pandemi Covid-19 yang mengakibatkan berbagai kegiatan bersama terpaksa dilakukan secara jarak jauh atau daring. Hal ini mendorong teknologi sebagai salah satu aspek komunikasi yang sangat penting, utamanya teknologi informasi. Salah satu kegiatan yang terimbas adalah kegiatan ajar mengajar. Pada awal-awal pandemi, sistem ajar mengajar memiliki banyak kekurangan dan hambatan dikarenakan belum siap terhadap perubahan langsung dan mendadak dari pembelajaran tatap muka menjadi pembelajaran jarak jauh. Namun, untuk persiapan untuk pembelajaran jarak jauh, telah dibuat dan dikembangkan suatu sistem yaitu *Learning Management System*.

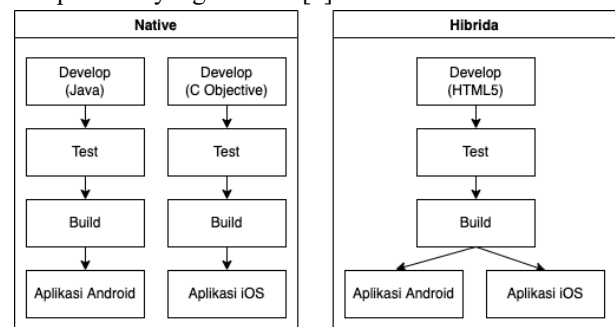
*Learning Management System* (LMS) merupakan suatu sistem yang mengelola sisi administrasi, dokumentasi, pelaporan, penyampaian, pembelajaran, dan pengawasan dalam proses ajar-mengajar. Suatu LMS dapat dibuat di luar kebutuhan institusi akademik dan membantu suatu perusahaan dalam mengelola layanan pembelajaran [1].

ITB telah mengembangkan suatu LMS bernama Edunex ITB. Sistem ini dibuat untuk mengganti dan memperbaharui

sistem LMS yang terdahulu digunakan ITB yang berbasis Moodle. Sistem ini memiliki berbagai fitur untuk mendukung civitas akademik, seperti penampilan materi pembelajaran, pemberian dan penilaian tugas dan ujian, sistem presensi kelas, dan notifikasi untuk tugas dan ujian. Untuk versi selanjutnya sudah ada rencana untuk membuat versi *mobile web* dan *mobile app* [2]. Hal ini diharapkan membantu membuat sistem Edunex lebih fleksibel dan menyesuaikan preferensi pengguna, dan mendukung sistem pembelajaran bauran (*hybrid*) yang ingin diadakan ITB dalam waktu yang mendatang.

## II. APLIKASI NATIVE

Aplikasi native merupakan aplikasi yang dibuat dan di-compile untuk suatu sistem operasi tertentu. Pada awal muncul suatu platform, pengembang hanya dapat membuat aplikasi secara native. Untuk membuat aplikasi ini, pengembang harus memakai bahasa dan alat yang berbeda untuk platform yang berbeda [3].



Gambar 1. Perbandingan Distribusi dan Upgrade Aplikasi Native dengan Hibrida

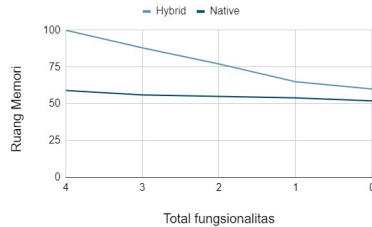
Aplikasi *native* dibuat menggunakan bahasa dan SDKs yang didukung oleh sistem aplikasi *smartphone* yang membuat aplikasi *native* memberikan performa yang lebih baik dan cepat. Hal ini dikarenakan API kerangka kerja aplikasi *native* mendukung pengaksesan fungsionalitas yang disediakan oleh *smartphone* secara langsung seperti kamera, GPS, dan aplikasi yang terpasang langsung di *smartphone* tersebut [QUE16].

TABEL I. PERBANDINGAN PERFORMA APLIKASI NATIVE DENGAN HIBRIDA

Kode	Aplikasi Hibrida	Aplikasi Native
Konsumsi waktu untuk instalasi (detik)	9.37	7.64
Konsumsi waktu untuk <i>Startup</i> aplikasi (detik)	1.58	1.11
Rasio penggunaan CPU (%)	11.76	5.54
Penggunaan memori (MB)	101.66	58.77

Kode	Aplikasi Hibrida	Aplikasi Native
Suhu baterai (°C)	45.32	37.54
Aliran jaringan (KB)	330.56	323.89

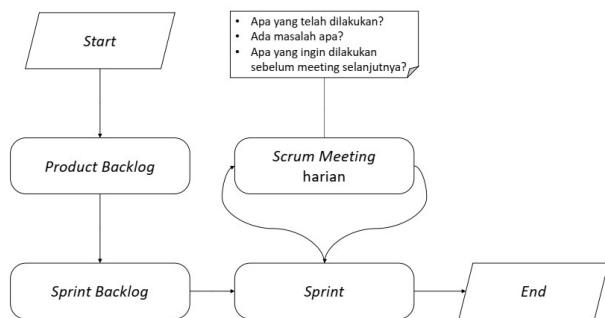
Selain itu, perbandingan performa juga dilakukan menggunakan uji vertikal yaitu dengan membandingkan konsumsi memori aplikasi jika ketika menjalankan aplikasi yang memiliki fungsionalitas perangkat seperti kamera, GPS, dan media [3].



Gambar 2. Perbandingan Distribusi dan Upgrade Aplikasi Native dengan Hibrida

### III. PROSES IMPLEMENTASI

Berdasarkan permasalahan, untuk meningkatkan kenyamanan pengguna Edunex ITB, terutama dalam hal fleksibilitas, maka pada Tugas Akhir ini akan dilakukan analisis dan perancangan aplikasi *native* Android. Metodologi yang digunakan dalam tugas akhir ini adalah menggunakan Agile Scrum.



Gambar 3. Laju Proses Scrum

Untuk penggunaan kakas, digunakan Figma untuk komunikasi prototipe aplikasi, dan Android Studio untuk pembuatan aplikasi secara *native*. Aplikasi akan dibangun dalam bahasa Kotlin.

#### A. Kebutuhan Sistem

Pada bagian ini, dipaparkan kebutuhan sistem yang terdiri dari fitur-fitur yang perlu dikembangkan, kebutuhan fungsional, dan kebutuhan non fungsional. Fitur-fitur yang perlu dikembangkan didapatkan dari tim Direktorat Pengembangan Pendidikan ITB.

TABEL II. FITUR APLIKASI EDUNEX YANG DIFOKUSKAN

Kode	Fitur	Persentase
FTR-1	Authentication	Sistem mampu menangani autentikasi dengan Edunex versi web
FTR-2	Authorization	Sistem mampu melakukan otorisasi pengguna
FTR-3	Course	Sistem mampu memberikan materi perkuliahan

Kode	Fitur	Persentase
FTR-4	Quiz/Exam	Sistem mampu menampilkan kuis atau ujian
FTR-5	Discussion	Sistem mampu melakukan penulisan jawaban kuis atau ujian dan mengumpulkannya

Dari tabel di atas, dilakukan penjabaran kebutuhan fungsional dan non fungsional dari aplikasi yang akan dibuat. Kebutuhan fungsional dan non fungsional tersebut dapat dilihat pada kedua tabel di bawah

TABEL III. KEBUTUHAN FUNGSIONAL

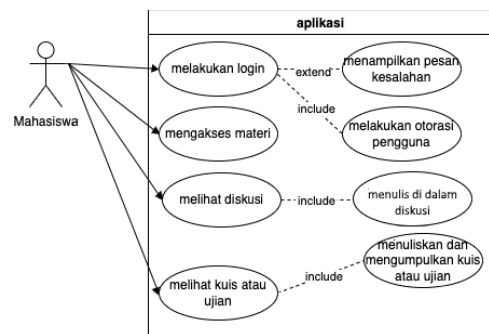
Kode	Penjelasan	Keterkaitan
FR-1	Sistem mampu menangani autentikasi dengan Edunex versi web	FTR-1
FR-2	Sistem mampu melakukan otorisasi pengguna	FTR-2
FR-3	Sistem mampu memberikan materi perkuliahan	FTR-3
FR-4	Sistem mampu menampilkan kuis atau ujian	FTR-4
FR-5	Sistem mampu melakukan penulisan jawaban kuis atau ujian dan mengumpulkannya	FTR-4
FR-6	Sistem dapat menuliskan dan mengikuti forum diskusi	FTR-5

TABEL IV. KEBUTUHAN NON FUNGSIONAL

Kode	Penjelasan	Keterkaitan
NFR-1	Sistem mampu menyimpan autentikasi dalam waktu yang lama	FTR-1, FR-1
NFR-2	Antarmuka aplikasi sederhana dan mudah dipahami	
NFR-3	Sistem dapat diakses kapan saja dan di mana saja	

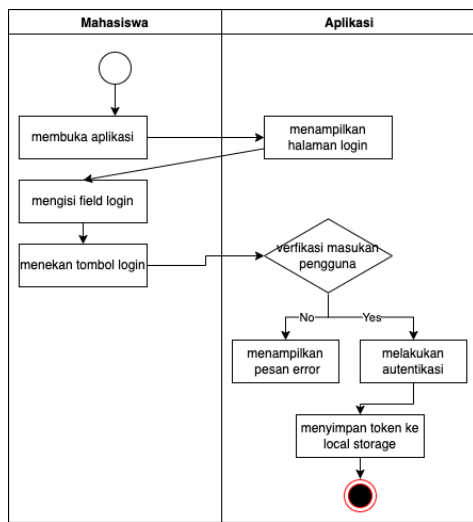
#### B. Pemodelan

Pada rancangan diagram *use case* di bawah, terdapat aktor utama yang merupakan pengguna mahasiswa. Mahasiswa dapat melakukan Log in, mengakses materi, melihat kuis atau ujian, membuka forum diskusi, dan menuliskan pesan diskusi.



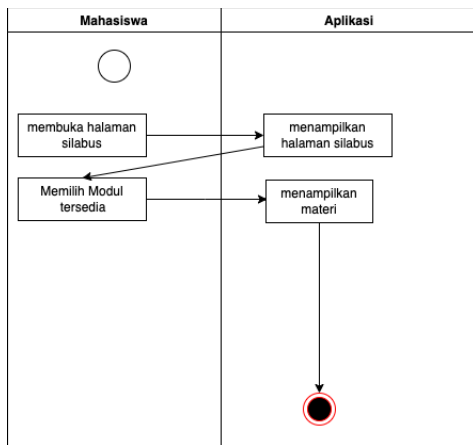
Gambar 4. Diagram Use Case

Di sini, pengguna mahasiswa didefinisikan sebagai Pengguna yang terdaftar sebagai Mahasiswa ITB atau yang memiliki credential akses sebagai mahasiswa



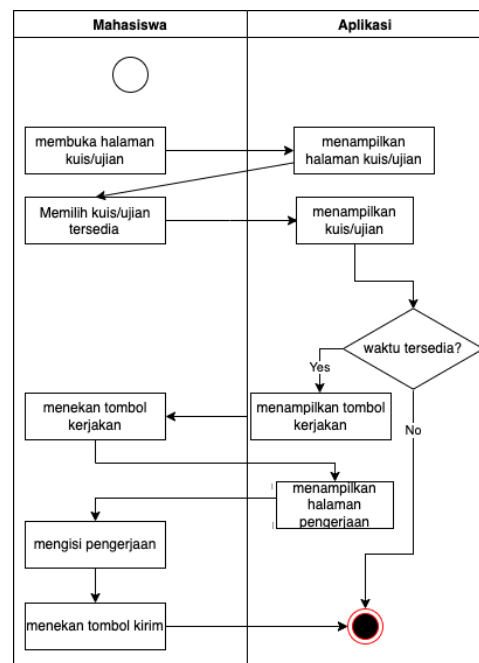
Gambar 5. Diagram Aktivitas Melakukan Log In

Skenario pertama yang dilakukan yaitu melakukan *log in* dengan memasukkan nama pengguna dan kata sandi pada ruang yang diberikan. Kemudian, pengguna menekan tombol masuk. Setelah itu, sistem akan melakukan autentikasi dan otorisasi untuk menentukan apakah pengguna telah *log in*.



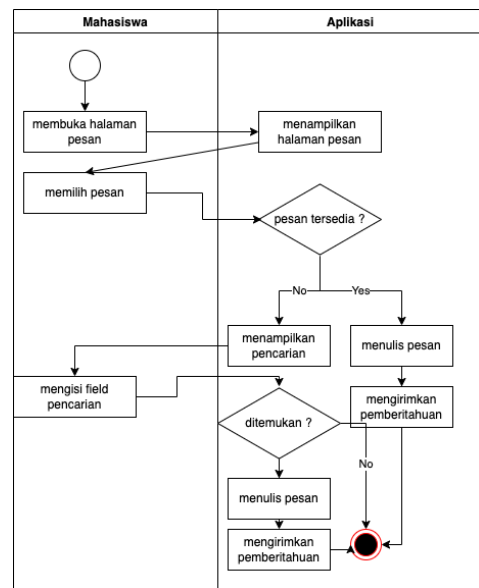
Gambar 6. Diagram Aktivitas Mengakses Bahan Ajar

Skenario kedua yaitu aktivitas mengakses bahan ajar. Kondisi awal untuk melakukan skenario ini yaitu pengguna telah terautentikasi. Kemudian pengguna masuk ke halaman silabus. Pada halaman ini akan ditampilkan modul yang tersedia.



Gambar 7. Diagram Aktivitas Mengakses Kuis atau Ujian

Pada skenario ini, pengguna dapat melakukan akses kuis atau ujian. Tampilan dibedakan untuk pengguna dosen dengan mahasiswa. Pada skenario ini pengguna dapat mengakses dan mengerjakan kuis atau ujian yang telah ada.



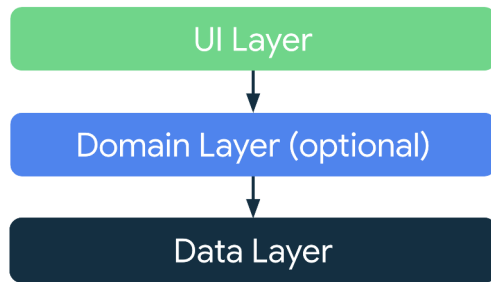
Gambar 8. Diagram Aktivitas Melakukan Pesan

Skenario keempat yaitu aktivitas melakukan pesan. Pengguna dapat membuka obrolan secara personal atau kelompok; Pada skenario ini tidak dibedakan antara pengguna dosen dengan pengguna mahasiswa. Setelah dibuat, pengguna dapat membuka halaman percakapan, dimana pengguna dapat mengirimkan dan membaca pesan yang diterima oleh lawan bicaranya.

### C. Arsitektur Aplikasi

Dalam aplikasi ini, akan digunakan arsitektur dasar yang disarankan oleh Android, yaitu UI-Domain-Data. Dalam arsitektur ini, aplikasi dapat dianggap memiliki setidaknya dua lapis, yaitu lapisan *User Interface* dan lapisan *Data*.

Lapisan *UI* dianggap di atas lapisan *Data*. Ada pula lapisan opsional di antara kedua lapisan tersebut, yaitu lapisan *Domain*.



Gambar 9. Diagram Arsitektur Aplikasi

Lapisan User Interface merupakan lapisan yang mengendalikan tampilan dan keluaran aplikasi seperti gambar, suara, teks, dll. Lapisan ini juga yang menerima dan memproses interaksi pengguna.

Lapisan Data merupakan lapisan yang mengendalikan data aplikasi, baik data luring sebagai internal database atau external file, maupun data luring yang dihasilkan dengan memanggil API atau situs web lainnya.

Lapisan Domain merupakan lapisan yang khusus apabila aplikasi memerlukan pemrosesan data lanjutan untuk memenuhi kebutuhan aplikasi. [4]

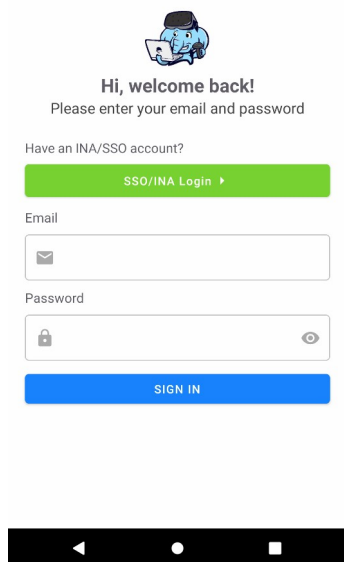
#### IV. HASIL

Proses implementasi aplikasi dilakukan dengan metode scrum. Scrum dilakukan dengan melakukan 11 sprint dengan durasi sprint 2 minggu. Setiap sprint dimulai dengan melakukan sprint planning untuk menentukan Product Backlog Item (PBI) yang akan dikerjakan. Setiap akhir sprint, dilakukan sprint retrospeksi untuk menilai keberjalanan dari sprint. Pada bagian ini, dijelaskan proses pengerjaan aplikasi dengan metode scrum yang dibagi setiap sprint.

TABEL V. *PRODUCT BACKLOG ITEM*

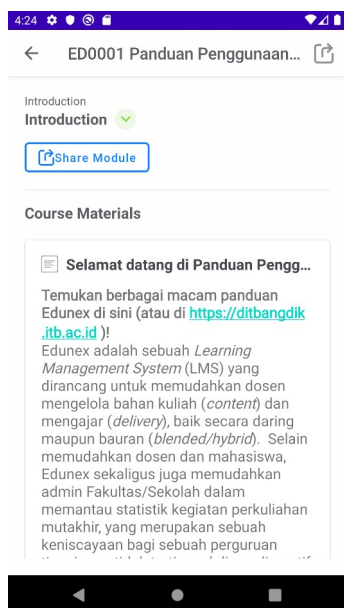
Kode	Product Backlog Item
1	layar <i>login</i> non-INA
	layar <i>login</i> INA
	Membuat layar utama aplikasi
	membuat boks mata kuliah
	membuat aksi di setiap boks mata kuliah
2	layar <i>login</i> non-INA
	layar <i>login</i> INA
	Membuat layar utama aplikasi
	Membuat boks-boks matakuliah di layar utama
	Membuat aksi navigasi di setiap boks matakuliah di layar utama
	Membuat layar isi matakuliah
	Menampilkan isi matakuliah
	Membuat halaman CRS
	Membuat boks akses CRS
	Membuat fitur pesan-memesan di dalam CRS

Kode	Product Backlog Item
	Membuat fitur <i>voting</i> di dalam CRS
3	layar <i>login</i> INA
	Membuat layar utama aplikasi
	Membuat boks-boks matakuliah di layar utama
	Membuat aksi navigasi di setiap boks matakuliah di layar utama
	Menampilkan isi matakuliah
	Membuat halaman CRS
	Membuat boks akses CRS
	Membuat fitur pesan-memesan di dalam CRS
	Membuat fitur <i>voting</i> di dalam CRS
4	Membuat halaman CRS
	Membuat boks akses CRS
	Membuat fitur pesan-memesan di dalam CRS
	Membuat fitur <i>voting</i> di dalam CRS
5	Membuat halaman CRS
	Membuat boks akses CRS
	Membuat fitur pesan-memesan di dalam CRS
	Membuat fitur <i>voting</i> di dalam CRS
6	Membuat fitur <i>voting</i> di dalam CRS
7	Membuat fitur <i>voting</i> di dalam CRS
8	Membuat fitur <i>voting</i> di dalam CRS
9	Membuat fitur <i>voting</i> di dalam CRS
10	-
11	Memperbaiki pengulangan dalam pemain video
	Memperbaiki peng-embed-an pesan <i>reply</i> di CRS
	Membuat layar ujian/kuis
	Membuat daftar pertanyaan di dalam ujian
	Menambahkan foto dosen dalam <i>course box</i>
	Memperbaiki tata letak tombol di layar utama, unduhan, kotak pesan, kirim pesan, dan masuk CRS
	Membuat pertanyaan pilihan ganda di dalam ujian
	Membuat pertanyaan pilihan banyak di dalam ujian
	Membuat pertanyaan esai di dalam ujian
	Membuat layar ujian tidak dapat keluar aplikasi
	Membuat fungsionalitas video loncat 10 detik ke depan dan belakang
	Membuat fungsionalitas dokumen dan video berlayar penuh
	Membuat ujian dapat di-submit



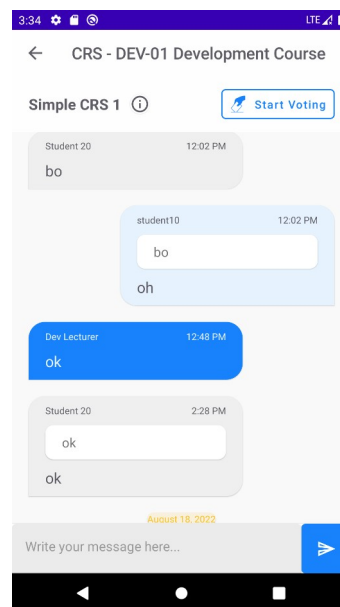
Gambar 10.

Tampilan *login*



Gambar 11.

Tampilan *courses*



Gambar 12.

Tampilan diskusi CRS

Setelah 11 sprint yang direncanakan, didapatkan total fitur-fitur yang telah diimplementasi dalam aplikasi sebagai berikut:

TABEL VI. TOTAL IMPLEMENTASI KEBUTUHAN FUNGSIONAL SISTEM

Kode	Penjelasan	Persentase
FR-1	Sistem mampu menangani autentikasi dengan Edunex versi web	100%
FR-2	Sistem mampu melakukan otorisasi pengguna	100%
FR-3	Sistem mampu memberikan materi perkuliahan	100%
FR-4	Sistem mampu menampilkan kuis atau ujian	40%
FR-5	Sistem mampu melakukan penulisan jawaban kuis atau ujian dan mengumpulkannya	0%
FR-6	Sistem dapat menuliskan dan mengikuti forum diskusi	100%

## V. KESIMPULAN

Perancangan aplikasi seluler Edunex dilakukan dengan tim 5 orang, dengan 2 orang mendesain *hi-fi* aplikasi, 2 orang membuat aplikasi, dan 1 orang yang menjaga kualitas dan kuantitas fitur aplikasi sesuai kebutuhan.

Pembangunan aplikasi seluler Edunex untuk versi Android dilakukan menggunakan metode *Scrum* dengan melakukan rencana 11 sprint. Setelah seluruh masa *sprint*, hasil dari aplikasi memenuhi 4,5 kebutuhan fungsional sistem yang direncanakan.

## REFERENSI

- [1] Foreman, Steven D., 2017, *The LMS Guidebook: Learning Management System*, ATD Press.
- [2] ---, 2021, *Buku Panduan Edunex 2021 Bagi Dosen*, Institut Teknologi Bandung.
- [3] Que, P, 2016, *A Comprehensive Comparison between Hybrid and Native App Paradigms*, 8th International Conference on Computational Intelligence and Communication Networks, doi:10.1109/cicn.2016.125.
- [4] ---, 2022, Guide to app architecture | Android Developers, URL: <https://developer.android.com/topic/architecture>. Waktu Akses: 07 Juli 2022.