

SKRIPSI

**RANCANG BANGUN APLIKASI PEMINJAMAN UANG “PINJAM AJA!”
DENGAN MODEL *P2P LENDING***



DISUSUN OLEH :

ALBERT KRISTAEN

DBC 113 008

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS PALANGKA RAYA

2020

SKRIPSI

**RANCANG BANGUN APLIKASI PEMINJAMAN UANG "PINJAM AJA!"
DENGAN MODEL P2P LENDING**

Sebagai salah satu syarat menyelesaikan Program Strata - 1
pada Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya

OLEH :

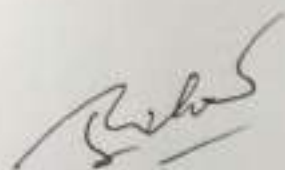
ALBERT KRISTAEN

NIM. DBC 113 008

Disetujui untuk diajukan dalam Seminar Akhir Skripsi

Palangka Raya, Juni 2020

Pembimbing I,



VIKTOR HANDRIANUS P., ST., MT
NIP. 19810606 200501 1 001

Pembimbing II,



PUTU BAGUS A.A.P., ST., M.Kom.
NIP. 19891022 201504 1 001

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PALANGKA RAYA**

2020

**RANCANG BANGUN APLIKASI PEMINJAMAN UANG "PINJAM AJA!" DENGAN
MODEL P2P LENDING**

SKRIPSI

Sebagai salah satu syarat untuk menyelesaikan Program Strata-1 pada Jurusan Teknik
Informatika Fakultas Teknik Universitas Palangka Raya

Oleh

ALBERT KRISTAEN

DBC 113 008

Telah dipertahankan didepan tim penguji, pada :

Hari/Tanggal : Selasa, 7 Juli 2020

Waktu : 09.00-10.30 WIB

1. FELICIA SYLVIANA, ST., MM
NIP. 19760118 200312 2 003 : (Ketua)
2. ABERTUN SAGIT SAHAY, ST., M.Eng
NIP. 19751212 200312 1 002 : (Anggota)
3. NOVA NOOR KAMALA SARI, ST., M.Kom
NIP. 19890407 201504 2 004 : (Anggota)
4. VIKTOR H. PRANATAWIJAYA, ST., MT
NIP. 19810606 200501 1 001 : (Anggota)
5. PUTU BAGUS A. A. P., ST., M. Kom
NIP. 19891022 201504 1 001 : (Anggota)

Mengetahui :

Fakultas Teknik
Universitas Palangka Raya
Dekan,

IZWALYU NUSWANTORO, M.T.
NIP. 19661119 199302 1 001

Jurusan / Program Studi Teknik Informatika
Fakultas Teknik Universitas Palangka Raya
Ketua Jurusan,

ABERTUN SAGIT SAHAY, S.T., M.Eng
NIP. 19751212 200312 1 002

PERNYATAAN

Dengan ini saya menyatakan dengan sebenar - benarnya bahwa dalam Skripsi ini tidak terdapat karya ilmiah yang pernah diajukan untuk memperoleh gelar kesarjanaan disuatu Perguruan Tinggi, serta tidak terdapat karya ilmiah atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam Skripsi ini dan disebutkan dalam Tinjauan Pustaka.

Palangka Raya, 13 Juli 2020



ALBERT KRISTAEN
DBC 113 008

RIWAYAT PENYUSUN

Data Diri

Nama : ALBERT KRISTAEN
NIM : DBC 113 008
Fakultas : Teknik
Jurusan/Program Studi : Teknik Informatika
Jenjang : Strata 1 (S-1)
Jenis Kelamin : LAKI-LAKI
Tempat, Tanggal Lahir : MUARA TEWEH, 3 APRIL 1995
Agama : KRISTEN
Status dalam Keluarga : Anak Kandung
Anak ke - : 1 (Satu)
Alamat : Jl. Sisingamangaraja I No. 9, Palangka Raya
No. Telpon/HP : +62 823 5465 8534



Data Orang Tua

Nama Ayah : ALIMAN
Pekerjaan Ayah : Petani
Nama Ibu : UJANG LUSIA
Pekerjaan Ibu : Petani
Alamat Orang Tua : Jl. Pertamina Rt.3, Desa Karang Langit, Barito Timur
No. Telpon/HP : +62 812 5117 9445

Riwayat Pendidikan *)

SD : SDN 1 Karang Langit (Tahun Lulus 2007)
SMP : SMPN 2 Tamiang Layang (Tahun Lulus 2010)
SMA : SMAN 1 Tamiang Layang (Tahun Lulus 2013)

Palangka Raya, Juli 2020

ALBERT KRISTAEN
DBC 113 008

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan kasih karunia penyertaan-Nya maka Laporan Skripsi ini dapat dikerjakan dengan baik sampai dengan selesai pada waktunya.

Adapun judul dari Skripsi yang dibuat yaitu, “*Rancang Bangun Aplikasi Peminjaman Uang “Pinjam Aja!” Dengan Model P2P Lending*”. Tujuan pengambilan judul ini adalah menghasilkan rancang bangun aplikasi Peminjaman Uang “Pinjam Aja!” dengan model *P2P Lending*.

Penulis juga menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam menyelesaikan Skripsi ini, khususnya kepada :

1. Tuhan Yesus Kristus yang telah mencurahkan Berkat-Nya kepada Penulis.
2. Orang Tua, Saudara dan Sahabat yang telah memberi semangat, motivasi dan doa kepada saya.
3. Bapak Viktor H. Pranata Wijaya, ST., MT dan Bapak Putu Bagus A. A. P., ST., M.Kom selaku dosen pembimbing, yang dengan sabar telah membimbing dan memberikan motivasi dalam pengerjaan laporan Skripsi dan aplikasinya hingga selesai.
4. Ibu Felicia Sylviana, ST., MT., Bapak Abertun Sagit Sahay, ST., M.Eng., dan Ibu Nova Noor Kamala Sari, ST., M.Kom selaku Dosen penguji yang telah banyak membantu memberikan masukan guna penyempurnaan aplikasi dan laporan Skripsi ini.
5. Rekan-rekan yang terlibat secara langsung maupun tidak langsung dalam pengerjaan proyek Skripsi ini.

Besar harapan saya, ke depannya semoga laporan Skripsi ini dapat bermanfaat bagi kita semua, terkhusus bagi semua mahasiswa Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya.

Palangka Raya, Juli 2020

Albert Kristaen

RANCANG BANGUN APLIKASI PEMINJAMAN UANG “PINJAM AJA!” DENGAN MODEL P2P LENDING

ALBERT KRISTAEN (DBC 113 008)

Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya
Kampus Tunjung Nyaho, Jln. Yos Sudarso, Palangka Raya 73112
Email : kristaen.albert@mhs.it.upr.ac.id

Ketika berbicara tentang meminjam uang, umumnya kita berpikir sebuah penyedia layanan keuangan seperti bank dengan persyaratan yang rumit dan membutuhkan proses panjang, bahkan membutuhkan jaminan untuk dapat melakukan peminjaman – karena bentuknya merupakan kredit dengan agunan. Lalu bagaimana kalau peminjam hanya membutuhkan dana tunai dengan nilai pinjaman kecil untuk keperluan mendesak, misalkan seorang mahasiswa yang membutuhkan uang untuk memperbaiki kerusakan kendaraannya namun pada saat yang bersamaan, dana yang dia miliki di tangan hanya cukup untuk pembayaran UKT semester. Dengan melihat contoh sederhana dari permasalahan keuangan tersebut, bagaimana jika saat membutuhkan uang pinjaman, prosesnya semudah menggunakan aplikasi belanja di sebuah *Marketplace*.

Pertanyaan singkat itu, menuntun penelitian kepada sebuah solusi untuk menjawab kondisi yang dipaparkan diatas, yaitu peminjaman uang dengan model P2P (*Peer-to-Peer*), yang secara singkat dapat dijelaskan sebagai sebuah *marketplace* yang mempertemukan peminjam dan pendana secara daring, dengan fokus pinjaman yang bersifat personal (Alexander Bachmann, et al., 2011). Pinjaman P2P tergolong platform baru dan inovatif untuk transaksi keuangan yang melewati perantara konvensional dengan menghubungkan langsung peminjam dan pemberi pinjaman (Shun Cai, et al, 2016). Metodologi pengembangan perangkat lunak yang digunakan dalam pembuatan proyek ini adalah metode *Waterfall* menurut Pressman (2012) yang terdiri dari tahapan Analisis, Desain Sistem, *Coding* & Pengujian, Implementasi dan Pemeliharaan. Desain sistem menggunakan UML dengan diagram yang digunakan diantaranya *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Bahasa pemrograman yang digunakan adalah Java pada sisi fungsi dan XML pada bagian tampilan, dengan menggunakan *IDE (Integrated Development Enviroment)* Android Studio 3.6. Sedangkan pada basis data menggunakan tipe NoSQL berbasis *Document* dari Google Firebase. *Blackbox testing* kemudian digunakan untuk menguji fungsional dari aplikasi apakah sudah berjalan sesuai dengan yang diharapkan.

Hasil dari penelitian diharapkan mampu menjadi salah satu model rancangan produk aplikasi peminjaman uang bertipe P2P *lending*, yang bertujuan untuk mempertemukan secara langsung peminjam dan pendana secara daring dalam sebuah *marketplace*.

Kata Kunci : P2P Lending, Android, Firebase, noSQL, waterfall

**DESIGN AND DEVELOPMENT OF MONEY LENDING APPLICATION
“PINJAM AJA!” WITH P2P LENDING MODEL**

ALBERT KRISTAEN (DBC 113 008)

Departement of Informatics Engineering Palangka Raya University
Tunjung Nyaho Campus, Yos Sudarso St., Palangka Raya 73112
Email : kristaen.albert@mhs.it.upr.ac.id

When talking about borrowing money, we generally think of a financial service provider like a bank with complicated requirements and requires a long process, even requiring collateral to be able to make loans - because the form is a loan with collateral. Then what if the borrower only needs cash with the value of a small loan for urgent needs, for example a student who needs money to repair damage to his vehicle but at the same time, the funds he has in hand are only enough for semester UKT payments. By seeing a simple example of these financial problems, what if when you need loan money, the process is as easy as using a shopping application in a Marketplace.

These short question, leading research into a solution to answer the conditions described above, namely lending money with the P2P model (Peer-to-Peer), which can be briefly explained as a marketplace that brings together borrowers and lenders online, with a focus on loans personal (Alexander Bachmann, et al., 2011). P2P loans are classified as a new and innovative platform for financial transactions that pass through conventional intermediaries by directly linking borrowers and lenders (Shun Cai, et al, 2016). The software development methodology used in making this project is the Waterfall method according to Pressman (2012) which consists of the stages of Analysis, System Design, Coding & Testing, Implementation and Maintenance. The system design uses UML with diagrams used including Use Case Diagrams, Activity Diagrams, and Class Diagrams. The programming language used is Java on the function side and XML on the display, using the IDE (Integrated Development Environment) of Android Studio 3.6. Whereas the database uses the Document-based NoSQL type from Google Firebase. Blackbox testing is then used to test the functionality of the application if it is running as expected.

The results of the study are expected to be one of the design models for P2P lending type of money lending application product, which aims to directly bring together borrowers and lenders online in a marketplace.

Keywords: P2P Lending, Android, Firebase, noSQL, waterfall

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	iv
HALAMAN RIWAYAT PENYUSUN	v
KATA PENGANTAR	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	2
1.5. Manfaat Penelitian.....	2
1.6. Sistematika Penulisan.....	3
1.7. Jadwal Kegiatan Penelitian.....	4
BAB II LANDASAN TEORI	5
2.1. Tinjauan Pustaka.....	5
2.2. Teori Pendukung.....	7
2.2.1. P2P <i>Lending</i>	7
2.2.1.1. Cara Kerja Layanan P2P <i>Lending</i>	8
2.2.1.2. Kelebihan Bagi Peminjam.....	9
2.2.1.3. Kekurangan Bagi Peminjam.....	10
2.2.1.4. Kelebihan Bagi Pendana.....	10

2.2.1.5. Kekurangan Bagi Pendana.....	11
2.2.1.6. Regulasi.....	11
2.2.2. Pengertian Rancang Bangun.....	15
2.2.3. Siklus Pengembangan Sistem (SDLC).....	15
2.2.4. UML (<i>Unified Modelling Language</i>).....	17
2.2.4.1. <i>Use Case Diagram</i>	18
2.2.4.2. <i>Class Diagram</i>	19
2.2.4.3. <i>Activity Diagram</i>	20
2.2.5. Android OS.....	20
2.2.6. NoSQL.....	21
2.2.7. Firebase.....	22
2.2.7.1. <i>Firebase Authentication</i>	22
2.2.7.2. <i>Firebase FireStore</i>	22
2.2.7.3. <i>Firebase Cloud Storage</i>	24
2.2.8. <i>Regex (Regular Expression)</i>	25
2.2.9. <i>Blackbox Testing</i>	26
BAB III METODOLOGI PENELITIAN	27
3.1. Metode Pengumpulan Data.....	27
3.1.1. Studi Pustaka.....	27
3.1.2. Studi Lapangan.....	27
3.1.3. Studi Literatur.....	27
3.2. Metode Pengembangan Sistem.....	28
3.2.1. <i>Analisis</i>	28
3.2.2. <i>System Design</i>	28
3.2.3. <i>Coding And Testing</i>	28
3.2.4. <i>Implementation</i>	29
3.2.5. <i>Maintenance</i>	29
3.3. Kerangka Penelitian.....	29
3.4. Peubah dan Pengukuran.....	31
3.5. Data Analisis Dari Aplikasi Yang Di Observasi.....	32
3.5.1. Observasi Bunga, Besaran Dan Syarat Pinjaman.....	32

3.5.2. Hasil Observasi Alur Pengajuan Pinjaman.....	33
3.5.3. Hasil Observasi Alur Pengajuan Pendanaan.....	34
3.6. Deskripsi Proyek.....	34
3.7. Perencanaan Proyek.....	34
3.7.1. Bunga, Besaran Dan Syarat Pinjaman.....	34
3.7.2. Peran/Aktor Dalam Proses Bisnis.....	34
3.7.3. Proses Bisnis Alur Pengajuan Pinjaman.....	35
3.7.4. Proses Bisnis Alur Pengajuan Pendanaan.....	36
3.7.5. Kalkulasi Simulasi Pinjaman.....	37
3.8. Desain Sistem.....	38
3.8.1. <i>Usecase Diagram</i>	38
3.8.2. <i>Activity Diagram</i>	39
3.8.3. <i>Class Diagram</i>	42
BAB IV HASIL DAN PEMBAHASAN.....	45
4.1. Kebutuhan Sistem.....	45
4.2. Implementasi Dan Evaluasi Sistem.....	46
4.2.1. Implementasi.....	46
4.2.1.1. Tampilan Sisi Pengguna.....	46
4.2.1.2. Tampilan Sisi Admin.....	55
4.2.2. Evaluasi Sistem.....	58
4.2.2.1. Aplikasi Sisi Pengguna.....	58
4.2.2.2. Aplikasi Sisi Admin.....	62
BAB V KESIMPULAN DAN SARAN.....	66
5.1. Kesimpulan.....	66
5.2. Saran.....	66
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
GAMBAR 2.1. <i>System Development Life Cycle</i> model <i>Waterfall</i>	16
GAMBAR 2.2. Contoh Diagram Model <i>Use Case</i>	19
GAMBAR 2.3. Contoh Model <i>Class Diagram</i>	19
GAMBAR 2.4. Contoh Model <i>Activity Diagram</i>	20
GAMBAR 2.5. Dokumen pada <i>Firestore</i>	23
GAMBAR 2.6. Koleksi dan Dokumen.....	24
GAMBAR 3.1. Metode Pengembangan Sistem <i>Waterfall</i> Pada Aplikasi “Pinjam Aja!”.....	29
GAMBAR 3.2. Kerangka Penelitian.....	30
GAMBAR 3.3. <i>Use Case Diagram</i>	38
GAMBAR 3.4. <i>Activity Diagram</i> Case Daftar/Login Pinjam Aja! Untuk <i>User</i> Peminjam Dan Pendana.....	39
GAMBAR 3.5. <i>Activity Diagram</i> Case <i>Dashboard</i> Peminjam.....	40
GAMBAR 3.6. <i>Activity Diagram</i> Case <i>Dashboard</i> Pendana.....	41
GAMBAR 3.7. <i>Activity Diagram</i> Case <i>Dashboard</i> Admin.....	42
GAMBAR 3.8. <i>Class Diagram</i> Dari Aplikasi Pinjam Aja!.....	43
GAMBAR 3.9. <i>Class Diagram</i> Dari Aplikasi PinjamAja! Admin.....	44
GAMBAR 4.1. <i>Splash Screen</i> aplikasi Pinjam Aja!.....	46
GAMBAR 4.2. <i>Activity</i> Selamat Datang.....	47
GAMBAR 4.3. Daftar Pengguna.....	47
GAMBAR 4.4. <i>Login</i> Pengguna.....	48
GAMBAR 4.5. Lupa Password.....	48
GAMBAR 4.6. <i>Dashboard</i> Peminjam opsi Beranda.....	49
GAMBAR 4.7. Dialog permintaan untuk melengkapi profil peminjam.....	49
GAMBAR 4.8. Pengisian Biodata Peminjam.....	50
GAMBAR 4.9. Permintaan Pinjaman.....	50
GAMBAR 4.10. Keterangan Pinjaman yang Aktif.....	51
GAMBAR 4.11. Transfer Cicilan.....	51

GAMBAR 4.12. <i>Dashboard</i> : Notifikasi.....	52
GAMBAR 4.13. <i>Dashboard</i> : Profil Peminjam.....	52
GAMBAR 4.14. Permintaan pendanaan pada beranda Pendana.....	53
GAMBAR 4.15. Pendanaan Pinjaman.....	53
GAMBAR 4.16. <i>Fragment</i> Dana dan Submenu detail Pinjaman.....	54
GAMBAR 4.17. <i>Dashboard User</i> : Profil Pendana.....	54
GAMBAR 4.18. <i>Activity Login</i> Admin.....	55
GAMBAR 4.19. <i>Fragment</i> Permintaan Pinjaman dan Submenu.....	55
GAMBAR 4.20. <i>Fragment</i> Pendanaan dan Submenu.....	56
GAMBAR 4.21. <i>Fragment</i> Cicilan beserta Submenu.....	57
GAMBAR 4.22. <i>Fragment</i> Admin.....	57

DAFTAR TABEL

	Halaman
TABEL 1.1. Jadwal Penelitian.....	4
TABEL 3.1. Peubah Dan Pengukuran.....	31
TABEL 3.2. Deskripsi Proyek.....	34
TABEL 4.1. Pengujian <i>Blackbox</i> , Daftar <i>User</i>	58
TABEL 4.2. Pengujian <i>Blackbox</i> , Login <i>User</i>	58
TABEL 4.3. Pengujian <i>Blackbox</i> , Lupa <i>password</i>	58
TABEL 4.4. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Peminjam : <i>Fragment</i> Beranda.....	59
TABEL 4.5. Pengujian <i>Blackbox</i> , Pengisian Biodata Peminjam.....	59
TABEL 4.6. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Peminjam : <i>Fragment</i> Rincian Pinjaman.....	60
TABEL 4.7. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Peminjam : <i>Fragment</i> Profil....	60
TABEL 4.8. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Pendana : <i>Fragment</i> Beranda..	61
TABEL 4.9. Pengujian <i>Blackbox</i> , Pengisian Biodata Peminjam.....	61
TABEL 4.10. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Pendana : <i>Fragment</i> Rincian Pendanaan.....	62
TABEL 4.11. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Pendana : <i>Fragment</i> Profil....	62
TABEL 4.12. Pengujian <i>Blackbox</i> , Login Admin.....	62
TABEL 4.13. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Admin : <i>Fragment</i> Permintaan Pinjaman.....	63
TABEL 4.14. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Admin : <i>Fragment</i> Pendanaan Pinjaman.....	63
Tabel 4.15. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Admin : <i>Fragment</i> Cicilan Pinjaman.....	64
TABEL 4.16. Pengujian <i>Blackbox</i> , <i>Dashboard</i> Admin : <i>Fragment</i> Admin.....	65

BAB I PENDAHULUAN

1.1. LATAR BELAKANG

Ketika berbicara tentang meminjam uang, umumnya kita berpikir sebuah penyedia layanan keuangan seperti bank dengan persyaratan yang rumit dan membutuhkan proses panjang, bahkan membutuhkan jaminan untuk dapat melakukan peminjaman – karena bentuknya merupakan kredit dengan agunan. Lalu bagaimana kalau peminjam hanya membutuhkan dana tunai dengan nilai pinjaman kecil untuk keperluan mendesak, misalkan seorang mahasiswa yang membutuhkan uang untuk memperbaiki kerusakan kendaraannya namun pada saat yang bersamaan, dana yang dia miliki di tangan hanya cukup untuk pembayaran UKT semester. Koperasi kelihatannya menjadi solusi yang tepat untuk persoalan ini. Namun pada praktiknya, koperasi pun membutuhkan anggota yang telah terdaftar dan memiliki simpanan pokok dalam akun koperasinya dan besar nilai pinjaman pun dipengaruhi oleh besarnya simpanan pokok, serta keaktifan dalam membayar iuran dalam keanggotaannya di Koperasi tersebut (Jonathan, 2018)

Melihat contoh kondisi di atas, respons tercepat yang ditangkap oleh pikiran kita, mungkin dengan meminjam dari orang terdekat atau kerabat, dengan langsung ke orang yang bersangkutan. Namun bagaimana jika orang terdekat Anda tidak memiliki solusi untuk itu, katakanlah sama-sama dalam keuangan yang sulit. Dengan melihat contoh sederhana dari permasalahan keuangan tersebut, bagaimana jika saat kita membutuhkan uang pinjaman, prosesnya semudah menggunakan aplikasi belanja di sebuah *Marketplace*.

Pertanyaan singkat itu, menuntun penelitian kepada sebuah solusi untuk menjawab kondisi yang dipaparkan diatas, yaitu peminjaman uang dengan model P2P (*Peer-to-Peer*), yang secara singkat dapat dijelaskan sebagai sebuah *marketplace* yang mempertemukan peminjam dan pendana secara daring, dengan fokus pinjaman yang bersifat personal (Alexander Bachmann, et al., 2011). Pinjaman P2P tergolong platform baru dan inovatif untuk transaksi keuangan yang

melewati perantara konvensional dengan menghubungkan langsung peminjam dan pemberi pinjaman (Shun Cai, et al, 2016).

Berdasarkan paparan latar belakang diatas, penulis tertarik untuk mengangkat Skripsi dengan judul : **“RANCANG BANGUN APLIKASI PEMINJAMAN UANG “PINJAM AJA!” DENGAN MODEL P2P LENDING”**

1.2. RUMUSAN MASALAH

Berdasarkan latar belakang yang diangkat oleh penulis, maka rumusan masalah dalam penelitian ini adalah : Bagaimana merancang dan membangun aplikasi pinjaman uang “Pinjam Aja!” dengan model *P2P Lending*?

1.3. BATASAN MASALAH

Berdasarkan perumusan masalah di atas, untuk menghindari meluasnya permasalahan, maka batasan masalah dari Rancang Bangun Aplikasi Pinjaman Uang “Pinjam Aja!” Dengan Model *P2P Lending* yang akan dibuat adalah sebagai berikut :

1. Fokus penelitian yang dilakukan oleh penulis hanya pada rancang bangun aplikasi pinjaman uang dengan model *P2P Lending*.
2. Pengujian aplikasi menggunakan metode *BlackBox Testing*.
3. Transaksi yang dilakukan antara ketiga aktor, yaitu admin, peminjam dan pendana dilakukan dalam mode simulasi.

1.4. TUJUAN PENELITIAN

Berdasarkan masalah yang ada, maka tujuan yang ingin dicapai oleh penulis adalah menghasilkan rancang bangun aplikasi Pinjaman Uang “Pinjam Aja!” dengan model *P2P Lending*.

1.5. MANFAAT PENELITIAN

Manfaat yang diharapkan dari pembuatan Rancang Bangun Aplikasi Pinjaman Uang “Pinjam Aja!” dengan model *P2P Lending* adalah :

1. Dari sisi peminjam, rancang bangun aplikasi yang dibuat diharapkan dapat membantu untuk mendapatkan dana kredit tanpa agunan dengan proses yang mudah dengan bunga ringan.
2. Dari sisi pemberi dana pinjaman, penerapan sistem yang berjalan dari rancang bangun aplikasi yang dibuat diharapkan dapat menjadi salah satu opsi sarana investasi jangka pendek.

1.6. SISTEMATIKA PENULISAN

BAB I PENDAHULUAN

Bab ini berisi pembahasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan dan jadwal kegiatan penelitian.

BAB II LANDASAN TEORI

Bab ini berisi tinjauan pustaka mengenai judul yang diangkat oleh penulis dalam judul skripsi dan paparan teori pendukung mengenai Pengertian Rancang Bangun, paparan mengenai *P2P lending*, dan beberapa teori pendukung lainnya seperti Siklus Pengembangan Aplikasi, Diagram UML dan *BlackBox Testing*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi paparan mengenai analisis dan desain dalam pembuatan rancang bangun aplikasi peminjaman uang “PinjamAja!” dengan model *P2P Lending*.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan mengenai hasil penelitian dari proses rancang bangun aplikasi yang dibuat oleh penulis. Hal yang dimaksud di antaranya kebutuhan sistem, implementasi dan evaluasi sistem dengan menggunakan *BlackBox Testing*.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi pembahasan mengenai kesimpulan dari hasil penelitian rancang bangun aplikasi peminjaman uang “PinjamAja!” dengan model *P2P Lending* dan saran mengenai pengembangan sistem selanjutnya dimasa yang akan datang.

BAB II LANDASAN TEORI

2.1. TINJAUAN PUSTAKA

Penelitian yang dilakukan oleh Helmi Fauzi Siregar dan Nilfa Sari pada tahun 2018 yang berjudul “Rancang Bangun Aplikasi Simpan Pinjam Uang Mahasiswa Fakultas Teknik Universitas Asahan Berbasis Web”, membahas tentang pembuatan aplikasi simpan pinjam yang berguna untuk mahasiswa, untuk mengurangi beban terkait masalah administrasi perkuliahan. Dari pembuatan aplikasi, *admin* dan *user* mengalami kemudahan dalam menangani masalah simpan pinjam uang yang dilakukan mahasiswa di universitas Asahan tersebut. Dari penelitian tersebut juga diberikan simpulan bagaimana, peminjaman berhasil menangani permasalahan nyata mahasiswa dalam masalah keuangan, yang jikalau tidak teratasi akan menyebabkan permasalahan dalam ujian MID maupun UAS hingga ada mahasiswa yang tidak mengikuti ujian.

Penelitian yang dilakukan oleh Shun Cai, Xi Lin, Di Xu dan Xin Fu pada tahun 2016 yang berjudul “*Judging online peer-to-peer lending behaviour: A comparison of first-time and repeated borrowing request*”, membahas tentang perilaku peminjam dana pada sebuah platform *P2P lending*, dengan membandingkan antara peminjam yang melakukan transaksi pinjaman untuk pertama kalinya dengan peminjam yang melakukan transaksi permintaan pinjaman kembali. Adapun pada penelitian yang dilakukan, bentuk pinjaman pada transaksi pinjam kembali dibagi dalam dua kategori, yaitu permintaan pinjaman berulang tanpa pernah memberikan pinjaman sebelumnya (*borrower without lending history*) dan pinjaman berulang dengan adanya sejarah pemberian pinjaman sebelumnya (*borrower with lending history*). Fokus penelitian yang dilakukan adalah untuk memberikan pemahaman tentang bagaimana penyediaan informasi dalam daftar pinjaman dapat mempengaruhi tingkat keberhasilan pendanaan dengan keadaan jenis pinjaman yang berbeda. Dari hasil penelitian dilakukan, bagi para peminjam pertama kali, keputusan pemberian pinjaman oleh pemodal/pemilik dana (*lender*)

berdasarkan pada informasi terbatas yang diberikan, evaluasi permintaan pinjaman berpatokan kepada besarnya suku bunga, besarnya permintaan pinjaman, jumlah verifikasi dan nilai kredit. Sedangkan bagi para peminjam yang meminta pinjaman kembali, sejarah pinjaman seperti tingkat keberhasilan permintaan pinjaman merupakan informasi penting yang dipertimbangkan oleh pendana dalam pengambilan keputusan. Selain itu pula, bagi para peminjam yang pernah menjadi pendana, rasio peminjaman-pendanaan (*borrow-lend ratio*) menjadi salah satu pertimbangan dalam suksesnya permintaan pinjaman yang dilakukan. Namun pada skenario ini, dampak dari sejarah peminjaman sebelumnya yang berhasil, tidak terlalu berpengaruh pada tingkat keberhasilan permintaan pinjaman baru.

Penelitian yang dilakukan oleh Ben-jiang Ma, Zheng-long Zhou dan Feng-ying Hu pada tahun 2017 dengan judul "*Pricing Mechanisms in the Online Peer-to-Peer Lending Market*", membahas tentang mekanisme penetapan suku bunga pinjaman dalam pasar daring P2P *lending*. Pada penelitian terdapat tiga klasifikasi mekanisme penetapan suku bunga yaitu *Borrower Pricing Mechanism* (BPM), *Auction Pricing Mechanism* (APM) dan *Posted-Pricing Mechanism* (PPM). Dari hasil penelitian yang dilakukan, model penetapan suku bunga pada BPM dan PPM yang paling menguntungkan bagi pihak pemberi pinjaman. Selanjutnya, harga (suku bunga) yang disediakan oleh pemberi pinjaman atau platform P2P dapat digunakan sebagai sinyal kualitas, dan menguntungkan dalam mengurangi asimetri informasi. Salah satu solusi yang mungkin adalah mengambil mekanisme harga tetap. Di bawah mekanisme seperti itu, pemberi pinjaman dapat mengajukan penawaran berdasarkan tingkat bunga yang ditawarkan oleh peminjam atau platform, dan tidak perlu bergantung pada perilaku penawaran pemberi pinjaman lainnya.

Pada penelitian yang dilakukan oleh Helmi Fauzi Siregar dan Nilfa Sari (2018), menampilkan keberhasilan pinjaman uang dalam mengatasi permasalahan keuangan mahasiswa dalam administrasi kampus. Namun sistem koperasi yang diterapkan, terasa tidak praktis ketika *user* membutuhkan pinjaman uang dengan proses yang cepat. Sehingga P2P *lending* yang berfokus kepada peminjaman uang saja dapat memangkas administrasi koperasi yang terasa lamban. Kemudian pada

penelitian yang dilakukan Shun Cai, et al (2016) memberikan gambaran tentang bagaimana permintaan pinjaman di proses dalam P2P *lending*, dimana proses peminjaman bagi peminjam pertama kali (*first-time borrower*) hanya membutuhkan informasi yang diberikan oleh peminjam, tidak perlu adanya bentuk simpanan pokok ataupun iuran bulanan sebagai indikator tingkat kepercayaan dalam pemberian pinjaman seperti pada koperasi. Namun, kelengkapan informasi pada platform P2P tersebut mempengaruhi besarnya tingkat kepercayaan pemberi pinjaman kepada peminjam. Selanjutnya, dalam penelitian yang dilakukan oleh Ben-jiang Ma, et al (2017) akan memberikan solusi bagaimana seharusnya sebuah platform P2P *lending* dibuat, yang menguntungkan bagi kedua belah pihak, yaitu dalam paparan pada bagian simpulan memberikan dua opsi terbaik dalam penentuan besaran suku bunga pinjaman, yaitu dengan model BPM, dimana besaran suku bunga ditawarkan oleh peminjam, atau bentuk model PPM, dimana besaran suku bunga pinjaman ditentukan oleh platform P2P.

2.2. TEORI PENDUKUNG

2.2.1. P2P Lending

P2P (*peer-to-peer*) *Lending* adalah penyelenggaraan layanan jasa keuangan untuk mempertemukan Pemberi Pinjaman dengan Penerima Pinjaman dalam rangka melakukan perjanjian pinjam meminjam melalui sistem elektronik dengan menggunakan jaringan internet. Layanan P2P merupakan penyelenggara badan hukum Indonesia yang menyediakan, mengelola, dan mengoperasikan Layanan Pinjam Meminjam Uang Berbasis Teknologi Informasi. Penerima Pinjaman (*borrower*) adalah orang dan/atau badan hukum yang mempunyai utang karena perjanjian Layanan Pinjam Meminjam Uang Berbasis Teknologi Informasi. Pemberi Pinjaman (Investor) adalah orang, badan hukum, dan/atau badan usaha yang mempunyai piutang karena perjanjian Layanan Pinjam Meminjam Uang Berbasis Teknologi Informasi. Peraturan soal P2P diatur dalam Peraturan OJK (POJK).

P2P lending, juga disebut sebagai "*debt crowdfunding*" atau "*markerplace lending*," muncul di Inggris pada tahun 2005. Platform pinjaman P2P pertama di

AS, Prosper, diluncurkan pada Februari 2006. Pinjaman P2P telah berkembang pesat, terutama setelah 2013. Saat ini, segmen terbesar dari pasar pinjaman P2P AS adalah pinjaman konsumen pribadi, dengan pinjaman lebih dari \$ 48 miliar berasal dari 3,2 juta peminjam pada tahun 2006 hingga 2018. Pinjaman P2P untuk konsumen didominasi oleh dua platform, LendingClub dan Prosper, dengan pasar saham masing-masing 84,9% dan 15,1%.

Pertumbuhan cepat pinjaman P2P baru-baru ini menarik, karena disintermediasi pinjaman melalui transaksi P2P tampaknya berlawanan dengan intuisi mengingat persepsi bahwa intermediasi keuangan itu sendiri muncul sebagai tanggapan terhadap ketidaksempurnaan pasar kredit (Diamond (1984); Boyd dan Prescott (1986)).

2.2.1.1. Cara Kerja Layanan P2P Lending

Karena terdapat dua pihak yang berperan dalam P2P *Lending*, yakni Peminjam dan Pendana/Investor, maka cara kerja P2P *Lending* bagi masing-masing dibedakan dan terpapar sebagai berikut:

a. Peminjam

Sebagai peminjam, yang perlu dilakukan adalah mengunggah semua dokumen yang dibutuhkan untuk mengajukan pinjaman secara *online* (yang relatif cepat prosesnya), yang di antaranya merupakan dokumen berisi laporan keuangan dalam jangka waktu tertentu dan juga tujuan peminjam dalam pinjaman tersebut. Bagi badan usaha yang mendaftar sebagai peminjam, biasanya diminta menyerahkan juga dokumen identitas, bukti legalitas perusahaan, dan laporan keuangan usaha.

Permohonan peminjaman peminjam bisa diterima atau pun ditolak, tentunya tergantung dari beragam faktor. Jika permohonan peminjam ditolak maka peminjam harus memperbaiki segala hal yang menjadi alasan penolakan permohonan peminjam.

Kemudian, jika diterima, suku bunga pinjaman akan diterapkan dan pengajuan pinjaman peminjam akan dimasukkan ke dalam *marketplace* yang tersedia agar semua pendana bisa melihat pengajuan pinjaman peminjam.

b. Pendana

Sebagai investor, nantinya pendana memiliki akses untuk menelusuri data-data pengajuan pinjaman di *dashboard* yang telah disediakan. Pendana juga pastinya bisa melihat semua data mengenai setiap pengajuan pinjaman, terutama data relevan mengenai si peminjam seperti pendapatan, tujuan pinjaman (bisnis, kesehatan, atau pendidikan) beserta alasannya.

Jika pendana memutuskan untuk menginvestasikan pinjaman tersebut, pendana bisa langsung menginvestasikan sejumlah dana setelah melakukan deposit sesuai tujuan investasi pendana.

Peminjam akan mencicil dana pinjamannya setiap bulan dan pendana akan mendapatkan keuntungan berupa pokok dan bunga. Besaran bunga akan tergantung pada suku bunga pinjaman yang diinvestasikan.

2.2.1.2. Kelebihan Bagi Peminjam

- a. Bagi si peminjam, salah satu manfaat terbesar dari *P2P Lending* adalah suku bunga yang rendah dibandingkan dengan suku bunga yang ditetapkan oleh lembaga keuangan resmi, misalnya, bank. Sebaliknya, pinjaman pribadi mungkin memiliki tingkat bunga antara 12-20% dari lembaga keuangan, itu pun masih lebih rendah ketimbang tagihan kartu kredit. Sedangkan pinjaman dari *P2P Lending* memiliki suku bunga yang lebih rendah.
- b. Kelebihan lainnya adalah proses pengajuan pinjamannya tidak seformal ketika mengajukan pinjaman di lembaga keuangan seperti bank. Prosesnya pun jauh lebih cepat dan mudah. Selain itu, peminjam tidak membutuhkan syarat-syarat sulit yang harus dipenuhi agar pinjaman disetujui.
- c. Nantinya, jika peminjam memiliki reputasi yang buruk soal pinjaman keuangan, peminjam bisa menjelaskan alasan di baliknya kenapa hal tersebut bisa terjadi. Selain itu, peminjam juga bisa mengajukan pinjaman untuk tujuan atau pun alasan apa pun selama ada orang yang akan menginvestasikan uangnya.
- d. Terakhir, *P2P Lending* merupakan pinjaman tanpa agunan yang artinya jaminan apa pun sama sekali tidak diperlukan.

2.2.1.3. Kekurangan Bagi Peminjam

- a. Suku bunga pinjaman *P2P Lending* melonjak naik saat kelayakan kredit peminjam jatuh.
- b. Jika peminjam telat membayar, tagihan akan sangat signifikan, di mana jika peminjam gagal membayar pinjaman peminjam, jumlah yang harus dibayar nantinya bisa melejit tinggi.
- c. Pinjaman hanya cocok untuk jangka pendek, sebab semakin lama jangka waktu pinjaman, tagihan akan terus naik. Ada kemungkinan bahwa kebutuhan dana pinjaman peminjam bisa terpenuhi secara keseluruhan, namun tidak ada jaminan bahwa seluruh pengajuan pinjaman dana akan terpenuhi.
- d. Misalnya peminjam membutuhkan pinjaman dana sebesar Rp 150 juta, dan jika hanya Rp 75 juta saja yang terpenuhi, pengajuan pinjaman peminjam berarti gagal dan dana yang sudah terkumpul akan dikembalikan ke para investor.

2.2.1.4. Kelebihan Bagi Pendana

- a. *P2P Lending* sudah resmi diatur dan diawasi oleh Otoritas Jasa Keuangan (OJK) lewat Peraturan OJK nomor 77/POJK.01/2016.
- b. Memberikan pinjaman sangat mudah dan cepat dengan *P2P Lending*, khususnya jika pendana memiliki dana lebih namun tidak tahu harus mengalokasikannya ke mana.
- c. Suku bunga pinjaman yang diterima memiliki nilai yang signifikan, sehingga lebih menguntungkan.
- d. Memberikan pinjaman melalui sistem *P2P Lending* ini juga memudahkan pendana untuk mendiversifikasi investasi pendana, sehingga memperbesar kesempatan untuk meraup keuntungan yang lebih besar.

2.2.1.5. Kekurangan Bagi Pendana

- a. Jika pendana menginvestasikan uang melalui *P2P Lending*, pendana tidak bisa menarik uang yang pendana investasikan kapan pun pendana mau, tidak seperti menyetor uang ke bank.
- b. Ada kemungkinan bahwa si peminjam akan gagal dalam mengembalikan uang pinjamannya, sehingga dana yang pendana pinjamkan bisa lenyap.
- c. Namun hal ini sudah diatasi oleh mayoritas platform *Peer-to-Peer Lending* dengan jaminan yang diberikan kepada pendana sebagai pendana. Risiko gagal bayar juga bisa diminimalisir dengan penerapan diversifikasi portofolio.

2.2.1.6. Regulasi

Sesuai POJK Nomor 77/POJK.01/2016, regulasi soal *P2P lending* adalah sebagai berikut:

a. Penyelenggara P2P

Badan hukum Penyelenggara berbentuk: a. perseroan terbatas; atau b. koperasi. Penyelenggara bisa didirikan dan dimiliki oleh Warga Negara/Badan Hukum Indonesia dan atau Asing. Penyelenggara wajib mengajukan pendaftaran dan perizinan kepada OJK.

- 1) Pendaftaran. Penyelenggara yang akan melakukan kegiatan P2P wajib mengajukan permohonan pendaftaran ke OJK. Sementara, penyelenggara yang sudah melakukan kegiatan P2P sebelum POJK 77 ini diberi waktu paling lama 6 bulan untuk mendaftar.
- 2) Perizinan. Penyelenggara yang telah terdaftar di OJK, wajib mengajukan permohonan izin sebagai Penyelenggara dalam jangka waktu paling lama 1 (satu) tahun sejak tanggal terdaftar di OJK. Jika dalam jangka waktu 1 tahun, Penyelenggara tidak mengajukan perizinan atau tidak memenuhi persyaratan perizinan, maka status terdaftar di OJK dinyatakan batal. Jika sudah batal status terdaftar, OJK menetapkan bahwa Penyelenggara tidak dapat lagi menyampaikan permohonan pendaftaran kepada OJK.

b. Modal Minimum

Modal minimum adalah Rp 1 Miliar saat pendaftaran dan Rp 2.5 Miliar pada saat mengajukan permohonan perizinan.

c. Kepemilikan

Penyelenggara oleh warga negara asing dan/atau badan hukum asing, baik secara langsung maupun tidak langsung paling banyak 85% (delapan puluh lima persen). Artinya, minimum kepemilikan Indonesia adalah 15% dalam layanan P2P. Perubahan kepemilikan Penyelenggara harus terlebih dahulu mendapatkan persetujuan dari OJK.

d. Kegiatan Usaha

Penyelenggara menyediakan, mengelola, dan mengoperasikan Layanan Pinjam Meminjam Uang Berbasis Teknologi Informasi dari pihak Pemberi Pinjaman kepada pihak Penerima Pinjaman yang sumber dananya berasal dari pihak Pemberi Pinjaman.

Penyelenggara wajib menggunakan pusat data dan pusat pemulihan bencana yang wajib ditempatkan di Indonesia. Wajib menyediakan rekam jejak audit (audit trail) terhadap seluruh kegiatannya di dalam sistem elektronik P2P.

Penyelenggara P2P dilarang untuk: bertindak sebagai Pemberi Pinjaman atau Penerima Pinjaman; menerbitkan surat utang; melakukan kegiatan usaha selain P2P.

e. Investor dan Peminjam

Peminjam di P2P harus berasal dan berdomisili di wilayah hukum Indonesia. Tetapi, investor (pemberi pinjaman) di P2P boleh berasal dari dalam dan luar negeri. Batas maksimum total pemberian pinjaman dana oleh P2P sebesar Rp 2 Milyar.

f. Kualifikasi Sumber Daya Manusia

Penyelenggara wajib memiliki sumber daya manusia yang memiliki keahlian dan/atau latar belakang di bidang teknologi informasi. Penyelenggara wajib memiliki paling sedikit 1 (satu) orang anggota Direksi dan 1 (satu) orang anggota Komisaris yang berpengalaman paling sedikit 1 (satu) tahun di industri jasa keuangan.

g. Mitigasi Risiko

Dalam proses pinjam meminjam, P2P wajib menggunakan akun *escrow* dan akun virtual. Penyelenggara P2P wajib menyediakan virtual akun bagi setiap Pemberi Pinjaman. Sementara dalam rangka pelunasan pinjaman, Penerima Pinjaman melakukan pembayaran melalui akun *escrow* penyelenggara untuk diteruskan ke virtual akun Pemberi Pinjaman.

h. Tanda Tangan Elektronik

Perjanjian dalam penyelenggaraan P2P dilaksanakan dengan menggunakan tanda tangan elektronik, sesuai dengan ketentuan peraturan perundang-undangan yang mengatur mengenai tanda tangan elektronik.

Aturan lebih lanjut mengenai biaya pinjaman, serta risiko yang ditanggung oleh pengguna LPMUBTI tercantum dalam “Pedoman Perilaku Pemberian Layanan Pinjam Meminjam Uang Berbasis Teknologi Informasi Secara Bertanggung Jawab”, yang disusun oleh AFPI sebagai asosiasi *Fintech Lending* di Indonesia yang ditunjuk secara resmi oleh OJK melalui surat S-5/D.05/2019 pada 17 Januari 2019. Berikut beberapa poin penting pokok-pokok pengaturan pelaksanaan LPMUBTI oleh AFPI :

- a. Adanya keterbukaan informasi biaya, dimana penyelenggara wajib mencantumkan seluruh biaya (*fees*) yang timbul dari pinjaman (*cost of borrowing*), termasuk biaya yang timbul di muka (*upfront fee*), bunga, biaya asuransi atau pertanggungan lain, provisi, biaya keterlambatan, dan biaya lainnya yang dikenakan kepada Penerima Pinjaman (Peminjam).
- b. Setiap penyelenggara wajib memberikan pernyataan (*disclaimer*) mengenai risiko yang ditanggung pemberi pinjaman (*lender*) di laman situs atau aplikasinya terkait hal-hal sebagai berikut :
 - 1) bahwa Pemberi Pinjaman sudah mengetahui dan menanggung sepenuhnya risiko atas pemberian pinjaman;
 - 2) bahwa tidak ada pihak yang menanggung risiko gagal bayar pinjaman maupun bertanggung jawab atas kerugian Penyelenggara, Pemberi Pinjaman dan/atau Penerima Pinjaman dari atau terkait dengan pemberian

- pinjaman atau kesepakatan apapun antara Penyelenggara, Pemberi Pinjaman dan/atau Penerima Pinjaman, kecuali pertanggung jawaban asuransi sesuai dengan syarat dan ketentuan yang berlaku;
- 3) bahwa Pemberi Pinjaman bertanggung-jawab atas risiko gagal bayar;
 - 4) bahwa Pemberi Pinjaman wajib mempelajari pengetahuan dasar mengenai LPMUBTI sebelum memberikan pinjaman;
 - 5) bahwa setiap kecurangan dan tindakan ilegal dilaporkan sepenuhnya kepada Otoritas Jasa Keuangan (OJK) dan masyarakat luas sesuai peraturan perundang-undangan yang berlaku; dan
 - 6) bahwa Pemberi Pinjaman sudah membaca dan mempelajari setiap syarat dan ketentuan.
- c. Setiap penyelenggara wajib memberikan pernyataan (*disclaimer*) mengenai risiko yang ditanggung penerima pinjaman (*borrower*) di laman situs atau aplikasinya terkait hal-hal sebagai berikut :
- 1) Bahwa Penerima Pinjaman sudah mengetahui dan menanggung sepenuhnya risiko atas pemberian pinjaman;
 - 2) bahwa Penerima Pinjaman sudah mengetahui risiko kehilangan aset ataupun harta kekayaan akibat gagal bayar;
 - 3) bahwa Penerima Pinjaman sudah mengetahui, mempertimbangkan dan menyetujui segala tingkat bunga dan biaya yang berlaku sesuai dengan kemampuan dalam melunasi pinjaman;
 - 4) bahwa Penerima Pinjaman diwajibkan untuk mempelajari pengetahuan dasar mengenai LPMUBTI sebelum memberikan pinjaman;
 - 5) bahwa setiap kecurangan dan tindakan ilegal dilaporkan sepenuhnya kepada Otoritas Jasa Keuangan (OJK) dan masyarakat luas sesuai peraturan perundang-undangan yang berlaku;
 - 6) bahwa catatan kredit Penerima Pinjaman akan dilaporkan secara berkala kepada otoritas jasa keuangan untuk kepentingan pusat Data *Fintech Lending* yang akan dimanfaatkan bersama dengan para pelaku industri perbankan nasional dan industri keuangan lainnya.

- 7) bahwa Penerima Pinjaman sudah membaca dan mempelajari setiap syarat dan ketentuan.
- d. Setiap penyelenggara wajib mempublikasikan kepada seluruh Pengguna terkait rekening dana penampungan (*virtual account* dan/atau *escrow account*) resmi yang digunakan dalam rangka fasilitasi pinjaman dan yang sudah di daftarkan kepada Otoritas Jasa Keuangan dan memastikan Pengguna tidak menggunakan rekening dana penampungan selain yang sudah dipublikasikan kepada pengguna.
- e. Penetapan total biaya pinjaman tidak melebihi suku bunga flat 0.8% per hari.
- f. Penetapan jumlah total biaya, biaya keterlambatan dan seluruh biaya-biaya lainnya sebesar maksimal 100% (seratus persen) dari nilai prinsipal pinjaman.

2.2.2. Pengertian Rancang Bangun

Kata “rancang” merupakan kata sifat dari “perancangan” yakni merupakan serangkaian prosedur untuk menerjemahkan hasil analisis dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen-komponen sistem diimplementasikan (Pressman, 2005).

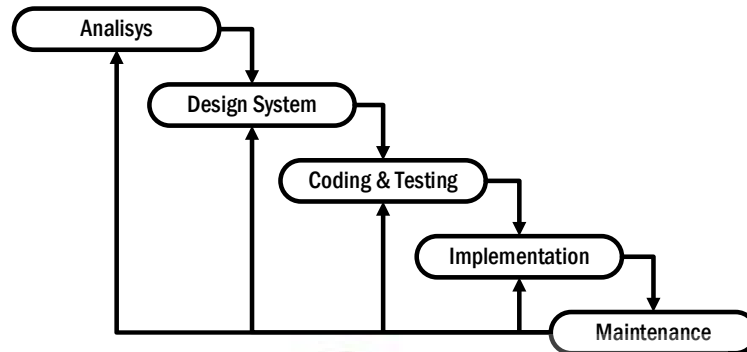
Kata “bangun” merupakan kata sifat dari “pembangunan” adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2005).

Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa kedalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut ataupun memperbaiki sistem yang sudah ada.

2.2.3. Siklus Pengembangan Sistem (SDLC)

Siklus pengembangan sistem adalah suatu pendekatan sistematis dan berurutan dalam pengembangan sebuah perangkat lunak. Metode *Waterfall* adalah suatu proses pengembangan perangkat lunak yang berurutan, dimana kemajuan dipandang sebagai sebuah proses yang terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, permodelan, implementasi (konstruksi) dan

pengujian. Berikut pada gambar 2.1. merupakan siklus pengembangan perangkat lunak berurutan/linear (Pressman, 2012).



GAMBAR 2.1. *System Development Life Cycle model Waterfall*

(Sumber : Pressman, 2012)

Dalam pengembangan model *Waterfall* memiliki beberapa tahapan yang runtut yaitu, *analysis, design system, coding and testing, implementation* dan *maintenance*. Berikut penjelasan tahapan-tahapan metode *Waterfall*.

a. *Analisis*

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Seorang sistem analis akan menggali informasi sebanyak-banyaknya dari pengguna, sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh pengguna tersebut. Tahapan ini akan menghasilkan dokumen kebutuhan pengguna atau bisa dikatakan sebagai data yang berhubungan dengan keinginan pengguna dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menerjemahkan ke dalam bahasa pemrograman.

b. *Design System*

Tahap desain merupakan tahapan untuk merancang desain basis data, antarmuka pengguna dan laporan dari proyek sistem informasi yang akan dibangun. Tahap ini merupakan tahap yang penting bagi pembuatan proyek sistem informasi. Hal ini dikarenakan tahapan ini akan menentukan berjalan tidaknya sebuah sistem dibuat.

c. *Coding and Testing*

Coding merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Proses ini dilakukan oleh *programmer* yang akan menerjemahkan transaksi yang diminta oleh pengguna. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan sebuah sistem. Setelah pengkodean selesai maka akan dilakukan pengujian terhadap sistem yang telah dibuat.

Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki. Tahapan ini bisa dikatakan final dalam pembuatan sebuah rancangan sebuah sistem.

d. *Implementation*

Implementasi merupakan tahapan untuk menerapkan sistem yang telah dibangun (penerapan rancang bangun program) guna memastikan proses bisnis yang ada saat ini. Pada tahap ini pengguna juga akan diberikan pelatihan guna menjalankan sistem yang baru. Seperti dijelaskan sebelumnya tahap ini tidak diperlukan dalam pengembangan rancang bangun sebuah perangkat lunak/sistem, karena bentuk pengetesan sebuah sistem merupakan simulasi skenario dari transaksi.

e. *Maintenance*

Tahap pemeliharaan merupakan tahapan dimana akan dilakukan perbaikan serta manajemen kembali sistem informasi yang telah diterapkan yang bertujuan guna pengembangan sistem. Tahap pemeliharaan juga bertujuan untuk menjaga kinerja sistem yang telah diterapkan agar dapat berjalan sesuai dan optimal. Tahap ini hanya dapat dilakukan pada sebuah aplikasi yang telah diterapkan atau diimplementasikan, dengan sistem berjalan bukan pada tahap pengguna yang disimulasikan.

2.2.4. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi

pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2005).

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh *Booch*, *Object Modeling Technique (OMT)* dan *Object Oriented Software Engineering (OOSE)*. Metode *Booch* dari Grady *Booch* sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan desain ke dalam 4 (empat) tahapan iteratif, yaitu: identifikasi kelas-kelas dan objek-objek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian antarmuka dan implementasi. Keunggulan metode *Booch* adalah pada detail dan kayanya dengan notasi dan elemen. Pemodelan *OMT* yang dikembangkan oleh *Rumbaugh* didasarkan pada analisis terstruktur dan pemodelan *entity-relationship* (Munawar, 2005).

2.2.4.1. Use Case Diagram

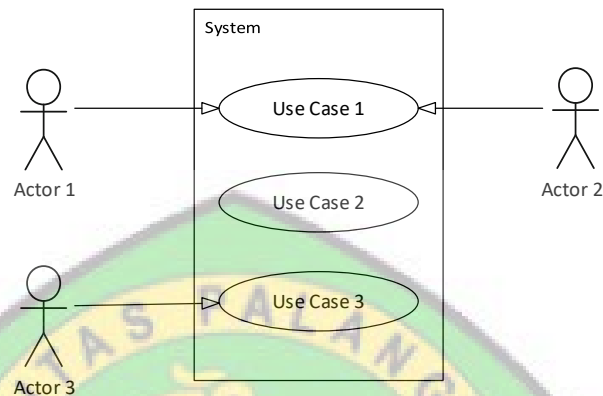
Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai (Munawar, 2005).

Dalam sebuah pembicaraan tentang *use case*, pengguna biasanya disebut dengan *actor*. *Actor* adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem. *Use case* adalah alat bantu terbaik guna menstimulasi pengguna potensial untuk mengatakan tentang suatu sistem dari sudut pandangnya. Diagram *use case* mempunyai 3 notasi yang menunjukkan aspek dari sistem (Munawar, 2005):

- a. *Actor* (Pengguna) yaitu abstraksi dari orang dan sistem lain yang mengaktifkan fungsi dari target sistem. *Actor* mewakili peran orang, sistem yang lain atau alat ketika berkomunikasi dengan *use case*.
- b. *Use Case* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* dibuat berdasarkan keperluan *actor*. *Use Case* harus merupakan “apa”

yang dikerjakan perangkat lunak, bukan “bagaimana” perangkat lunak mengerjakannya. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor*.

- c. *Relationship* (hubungan) yaitu hubungan antara *actor*/pelaku dengan *use case* dimana terjadi interaksi di antara mereka.

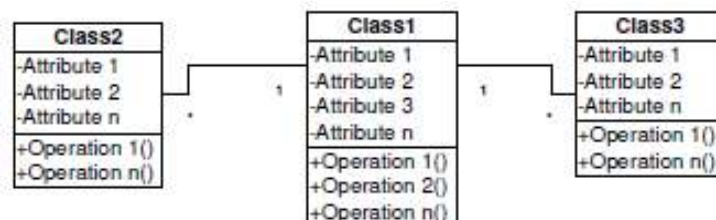


GAMBAR 2.2. Contoh Diagram Model Use Case (Munawar, 2005)

2.2.4.2. Class Diagram

Class dalam notasi UML digambarkan dengan kotak. Nama *class* menggunakan huruf besar di awal kalimatnya dan diletakkan diatas kotak. Bila *class* mempunyai nama yang terdiri dari 2 (dua) suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar. *Attribute* adalah *property* dari sebuah *class*. *Attribute* ini melukiskan batas nilai yang mungkin ada pada obyek dari *class*. Sebuah *class* mungkin mempunyai nol atau lebih *attribute* (Munawar, 2005).

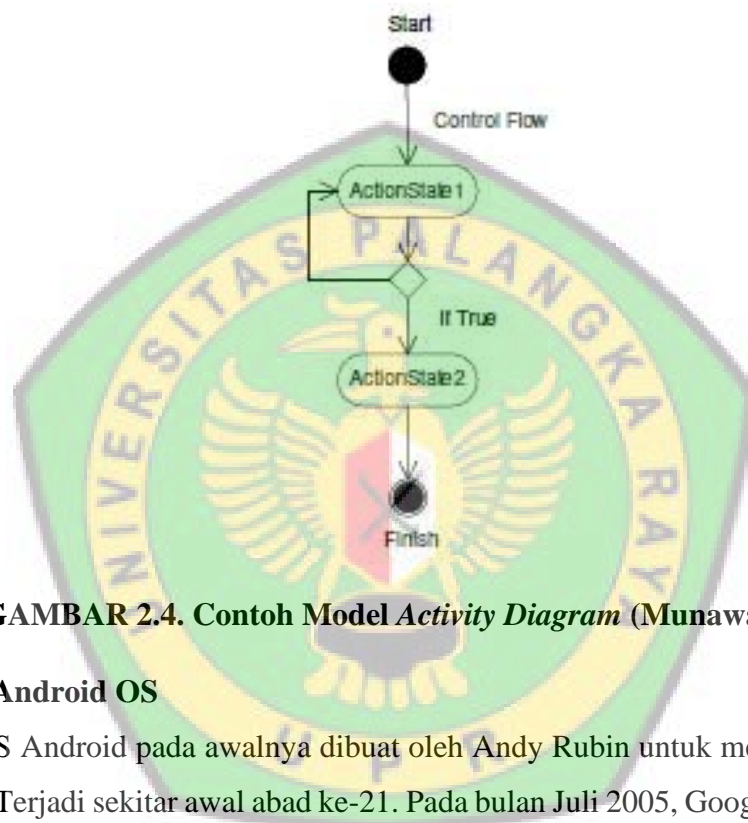
Operation adalah sesuatu yang bisa dilakukan oleh sebuah *class* atau yang anda (atau *class* yang lain) dapat lakukan untuk sebuah *class*. *Responsibility* adalah keterangan tentang apa yang akan dilakukan *class* yaitu apa yang akan dicapai oleh *attribute* dan *operation* (Munawar, 2005).



GAMBAR 2.3. Contoh Model Class Diagram (Munawar, 2005)

2.2.4.3. Activity Diagram

Activity Diagram adalah teknik untuk mendeskripsikan logika *procedural*, proses bisnis dan aliran kerja dalam banyak kasus. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku parallel sedangkan *flowchart* tidak bisa (Munawar, 2005).



GAMBAR 2.4. Contoh Model Activity Diagram (Munawar, 2005)

2.2.5. Android OS

OS Android pada awalnya dibuat oleh Andy Rubin untuk menjadi OS untuk ponsel. Terjadi sekitar awal abad ke-21. Pada bulan Juli 2005, Google mengakuisisi Android dan menjadikan Andy Rubin sebagai Wakil Presiden Senior Platform Mobile untuk Google, di mana ia bertahan hingga November 2014. Banyak yang merasa akuisisi OS Android oleh Google ini sebagian besar sebagai respons terhadap penampilan iPhone Apple di sekitar waktu yang sama. Namun, ada cukup banyak pemain besar lainnya, seperti RIM Blackberry, Nokia Symbian, dan Microsoft Windows Mobile, yang dianggap sebagai keputusan bisnis yang cerdas bagi Google untuk membeli bakat teknik dari Android Incorporated bersama dengan kekayaan intelektual OS Android. Ini memungkinkan Google untuk memasukkan perusahaan mesin pencari Internet mereka ke pasar ponsel yang

sedang berkembang, yang sekarang banyak orang sebut sebagai Internet 2.0 (Wallace Jackson, 2017:1).

Internet 2.0, atau Internet Seluler, memungkinkan pengguna produk elektronik konsumen untuk mengakses konten melalui jaringan data yang sangat beragam, menggunakan perangkat elektronik konsumen portabel (Wallace Jackson, 2017:2).

Dalam penulisan karya ilmiah ini penulis sendiri menggunakan Android 7.1.2 sebagai *unit test* dan versi SDK yang digunakan dalam pengembangan aplikasi.

2.2.6. NoSQL

Basis data NoSQL (awalnya merujuk ke "non SQL" atau "non relational") menyediakan mekanisme untuk penyimpanan dan pengambilan data yang dimodelkan dalam cara selain hubungan tabular yang digunakan dalam database relasional. Database semacam itu sudah ada sejak akhir 1960-an, tetapi nama "NoSQL" baru diciptakan pada awal abad ke-21, dipicu oleh kebutuhan perusahaan Web 2.0 (C. Mohan, 2013). Basis data NoSQL semakin banyak digunakan dalam data besar dan *real-time web application*. Sistem NoSQL juga kadang-kadang disebut "*Not Only SQL*" untuk menekankan bahwa mereka dapat mendukung bahasa query seperti SQL, atau duduk berdampingan dengan database SQL.

Motivasi untuk pendekatan ini meliputi: kesederhanaan desain, penskalaan "horizontal" yang lebih sederhana untuk kelompok mesin (yang merupakan masalah bagi basis data relasional) (Neal Leavitt, 2010), kontrol yang lebih baik atas ketersediaan dan membatasi ketidakcocokan impedansi objek-relasional. Struktur data yang digunakan oleh basis data NoSQL (mis. Nilai kunci, kolom lebar, grafik, atau dokumen) berbeda dari yang digunakan secara default di basis data relasional, membuat beberapa operasi lebih cepat di NoSQL. Kesesuaian khusus dari basis data NoSQL yang diberikan tergantung pada masalah yang harus dipecahkan. Kadang-kadang struktur data yang digunakan oleh database NoSQL juga dipandang sebagai "lebih fleksibel" daripada tabel database relasional.

2.2.7. Firebase

Pada bagian *server-side (back-end)* aplikasi Android dibuat penulis menggunakan layanan Firebase dari Google, Inc, untuk mengakomodasi keperluan Autentikasi, Basis Data dan Penyimpanan Awan (Cloud Storage) pada aplikasi.

2.2.7.1. Firebase Authentication

Fitur Firebase *Authentication* digunakan untuk mengetahui identitas pengguna untuk dapat memberikan data khusus yang hanya diberikan kepada pengguna tersebut, sehingga formulir *login* yang aman dibutuhkan untuk proses autentikasi ini. Autentikasi yang dilakukan terintegrasi dengan layanan Firebase *FireStore* (Basis Data) dan *Cloud Storage*, sehingga ada sistem kontrol terhadap hubungan antara *user* dan data apa saja yang dapat diakses oleh *user* tersebut (Laurence Moroney, 2017:3)

Proses pendaftaran akun baru pengguna, dimulai dengan memanggil *method* `createUserWithEmailAndPassword()`, *method* ini secara otomatis akan *generate Unique User ID (UUID)* yang digunakan sebagai tautan fungsi yang menghubungkan ketiga layanan utama Firebase, yaitu *Authentication* (Autentikasi), *FireStore (Database)* dan *Cloud Storage* (Penyimpanan Awan).

2.2.7.2. Firebase FireStore

Cloud Firestore adalah *database* NoSQL berorientasi dokumen. Tidak seperti *database SQL*, *database* tersebut tidak memiliki tabel atau baris. Sebagai gantinya, Anda menyimpan data dalam dokumen, yang disusun menjadi koleksi.

Setiap dokumen berisi kumpulan pasangan kunci/nilai. *Cloud Firestore* dioptimalkan untuk menyimpan koleksi besar berbagai dokumen kecil. Semua dokumen harus disimpan dalam koleksi. Dokumen dapat berisi sub koleksi dan objek bertingkat, yang dapat menyertakan kolom standar seperti *string* atau objek kompleks seperti daftar. Koleksi dan dokumen dibuat secara implisit di *Cloud Firestore*. Cukup tetapkan data ke dokumen dalam koleksi. Jika koleksi atau dokumen tidak ada, *Cloud Firestore* akan membuatnya.

Di *Cloud Firestore*, unit penyimpanan adalah dokumen. Dokumen adalah data ringan yang berisi kolom, yang dipetakan ke nilai. Setiap dokumen diidentifikasi dengan nama. Dokumen yang mewakili pengguna *alovelace* mungkin terlihat seperti pada gambar 2.5.a. Objek kompleks dan bertingkat dalam dokumen disebut peta (*map*). Misalnya, pengembang dapat membuat struktur nama pengguna dari contoh di atas dengan peta seperti pada gambar 2.5.b.



GAMBAR 2.5. Dokumen pada *Firestore*

Dokumen tersebut mungkin sangat mirip dengan JSON. Pada dasarnya, dokumen tersebut memang JSON. Ada beberapa perbedaan (misalnya, dokumen mendukung jenis data tambahan dan ukurannya terbatas hingga 1 MB). Namun secara umum, pengembang dapat memperlakukan dokumen sebagai data JSON ringan.

Dokumen berada dalam koleksi, yang merupakan wadahnya. Misalnya, aplikasi dapat memiliki koleksi *users* untuk menampung berbagai pengguna, masing-masing diwakili oleh dokumen, seperti terlihat pada gambar 2.6. berikut



GAMBAR 2.6. Koleksi dan Dokumen

Cloud Firestore tidak berskema, sehingga pengembang bebas memilih kolom apa yang akan dimasukkan ke dalam setiap dokumen, dan jenis data apa yang akan

disimpan pada kolom tersebut. Dokumen dalam koleksi yang sama dapat berisi berbagai kolom, atau menyimpan berbagai jenis data di kolom tersebut. Namun, sebaiknya gunakan kolom dan jenis data yang sama di beberapa dokumen, sehingga pembuatan kueri dokumen dapat lebih mudah.

Koleksi hanya berisi dokumen. Koleksi tidak dapat langsung berisi kolom mentah dengan nilai, dan tidak dapat berisi koleksi lainnya. Nama dokumen dalam koleksi bersifat unik. Pengembang dapat memasukkan kuncinya (*key*) sendiri, seperti ID pengguna, atau pengembang dapat membiarkan *Cloud Firestore* membuat ID acak untuk Anda secara otomatis.

Pengembang tidak perlu "membuat" atau "menghapus" koleksi. Setelah membuat dokumen pertama dalam koleksi, koleksi tersebut akan tersedia. Jika menghapus semua dokumen dalam koleksi, koleksi tersebut tidak akan tersedia lagi.

2.2.7.3. **Firestore Cloud Storage**

Developer menggunakan *Firestore SDK* untuk *Cloud Storage* untuk mengupload dan mendownload file langsung dari klien. Jika koneksi jaringan buruk, klien bisa mencoba operasi ini lagi dari posisi terakhir saat pengoperasian terhenti, sehingga menghemat waktu dan bandwidth pengguna.

Cloud Storage menyimpan *file* di keranjang *Google Cloud Storage*, sehingga membuatnya mudah diakses melalui *Firestore* dan *Google Cloud*. Dengan begitu, pengembang memiliki fleksibilitas untuk mengunggah dan mengunduh *file* dari klien seluler melalui *Firestore SDK*, dan melakukan pemrosesan sisi server seperti pemfilteran gambar atau *transcoding* video menggunakan *Google Cloud Platform*. *Cloud Storage* akan diskalakan secara otomatis, yang berarti tidak perlu bermigrasi ke penyedia lain.

Firestore SDK untuk *Cloud Storage* terintegrasi sempurna dengan *Firestore Authentication* untuk mengidentifikasi pengguna, dan *Firestore* menyediakan bahasa keamanan deklaratif yang dapat digunakan untuk menyetel kontrol akses pada masing-masing *file* atau kumpulan *file*, sehingga pengembang bisa menyetelnya sebagai publik atau pribadi sesuai keinginan.

2.2.8. *Regex (Regular Expression)*

Regular Expression adalah *string* teks disandikan khusus (*specially encoded*) yang digunakan sebagai pola untuk mencocokkan set *string*. *Regex* mulai muncul pada tahun 1940-an sebagai cara untuk menggambarkan bahasa reguler, tetapi benar-benar mulai muncul di dunia pemrograman pada tahun 1970-an. Tempat pertama yang ditemukan muncul di editor teks QED yang ditulis oleh Ken Thompson. Berikut kutipan verbatim definisi *regular expression* menurut Ken Thompson dalam Buku Michael Fitzgerald yang berjudul “*Introducing Regular Expression*” :

“*A regular expression is a pattern which specifies a set of strings of characters; it is said to match certain strings.*” - Ken Thompson. (Michael Fitzgerald, 2012:1)

Ekspresi reguler memiliki reputasi sulit untuk dikuasai, tetapi itu semua tergantung pada bagaimana *regex* diimplementasikan. Ekspresi reguler adalah tentang mencocokkan dan menemukan pola dalam teks, dari pola sederhana hingga yang sangat kompleks (Michael Fitzgerald, 2012:1)

Berikut merupakan beberapa pola *regex* yang dibuat oleh penulis dalam aplikasi untuk memvalidasi *input* registrasi *user* :

a. Validasi Nama

$[a-zA-Z\s]^+$: Pola tersebut digunakan untuk memvalidasi *input* yang diterima pada kolom nama hanya berupa karakter a-z, Z-A dan spasi saja.

b. Validasi Email

$([a-zA-Z0-9._ -]^+){3,}@([a-z.-]^+){3,}\.([a-z]^+){2,}$: Pola tersebut digunakan untuk memvalidasi *input*-an yang diterima pada kolom email, dengan mengikuti pola `username@mail_server.domain`.

Username yang valid diterima berupa karakter a-z, A-Z, 0-9, titik (.), *underscore* (_) dan *dash* (-) dengan minimal 3 karakter, kemudian diikuti dengan karakter @. Selanjutnya pada *mail_server* karakter yang diperbolehkan adalah a-z, titik (.) dan *dash* (-) dengan minimal total 3 karakter, kemudian diikuti dengan tanda titik (.). Selanjutnya pada domain, karakter yang diperbolehkan adalah a-z dengan minimal total 2 karakter.

c. Validasi Nomor Telepon

[+62][0-9]+ : Pola tersebut digunakan untuk memvalidasi *input* yang diterima aplikasi pada kolom nomor telepon, dengan mengikuti pola kode negara diikuti nomor telepon. [+62] menunjukkan bahwa 3 karakter awal yang valid adalah +62 yang menunjukkan kode negara nomor telepon Indonesia, diikuti dengan set angka 0-9 yang dapat muncul lebih dari satu kali (simbol + dalam *regex*).

2.2.9. Blackbox Testing

Dalam *testing* dan implementasi sistem dikenal 2 metode pengujian yang populer, yakni pengujian *black-box* dan pengujian *white-box*. *Black-Box Testing* merupakan pengujian berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program (Pranto Busono, 2002).

Berikut ini merupakan ciri-ciri *Black-Box Testing*, menurut Pranto Busono :

- a. *Black-Box Testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan spesifikasi kebutuhan dari *software*.
- b. *Black-Box Testing* bukan merupakan teknik alternatif daripada *white-box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup kesalahan (*error*) dengan kelas yang berbeda dari metode *white-box testing*.
- c. *Black-Box Testing* melakukan pengujian tanpa pengetahuan detail struktur internal dari sistem atau komponen yang dites. Juga disebut sebagai *behavioral testing*, *spesification-based testing*, *input/output testing* atau *functional testing*.
- d. Kategori kesalahan yang akan diketahui melalui *black-box testing*, antara lain : Fungsi yang hilang atau tak benar, kesalahan dari antar-muka (*interface*), kesalahan dari struktur data atau akses eksternal database, kesalahan dari kinerja atau tingkah laku (*behaviour*) dan kesalahan dari inisialisasi dan terminasi.

BAB III

METODOLOGI PENELITIAN

3.1. METODE PENGUMPULAN DATA

Metode pengumpulan data yang akan digunakan dalam melakukan penelitian ini dibagi menjadi tiga, yaitu studi pustaka, studi lapangan dan studi literatur.

3.1.1. Studi Pustaka

Studi pustaka yang dilakukan dalam penelitian ini dengan cara melihat, membaca, mencatat dan mengutip kepada suatu hasil penulisan penelitian yang sudah dicetak dan dijadikan referensi data dalam pemecahan suatu masalah yang ada pada Rancang Bangun Aplikasi Peminjaman Uang “Pinjam Aja!” Dengan Model P2P *Lending*. Bentuk dari hasil penelitian dan penulisan bisa berbentuk buku kamus, buku ensiklopedia jurnal, *yearbook*, *abstract journal*, *review journal*, *periodical*, *circular*, *leafet* dan buletin.

3.1.2. Studi Lapangan

Metode penelitian lapangan yang dilakukan adalah observasi. Observasi yang dilakukan nantinya dengan melihat contoh dari aplikasi P2P *lending* yang sudah memiliki izin OJK yang terdapat di *PlayStore*. Adapun aplikasi yang di observasi adalah TunaiKita dan PinjamanGo.

3.1.3. Studi Literatur

Studi literatur dalam penelitian ini dilakukan untuk menambah referensi teori-teori yang digunakan dalam penelitian, serta bertujuan untuk mempelajari literatur yang dapat mendukung penelitian terhadap Rancang Bangun Aplikasi Peminjaman Uang “Pinjam Aja!” Dengan Model P2P *Lending*. Sumber-sumber yang dapat dijadikan sumber literatur antara lain skripsi sejenis, tesis dan jurnal-jurnal terkait dengan penelitian yang akan diteliti.

3.2. METODE PENGEMBANGAN SISTEM

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode pengembangan sistem *waterfall* dengan permodelan objek secara bertahap dimulai dari *analysis, design system, coding and testing, implementation* dan *maintenance*. Namun tahapan *maintenance* tidak dilakukan karena aplikasi yang dibuat sampai kepada bentuk rancang bangun.

Pengembangan sistem menggunakan notasi UML (*Unified Modelling Language*). Adapun diagram yang digunakan dalam penelitian ini antara lain: *Use Case Diagram, Activity Diagram* dan *Class Diagram*.

3.2.1. *Analisis*

Dalam tahap ini, terdapat beberapa langkah yang perlu dilakukan dalam pembuatan sebuah rancang bangun aplikasi peminjaman uang “Pinjam Aja!”, antara lain :

- a. Identifikasi alur bisnis dalam *P2P lending online*, baik dalam proses permintaan pinjaman maupun proses pemberian pinjaman.
- b. Menentukan batasan ruang lingkup sistem yang akan dibangun.
- c. Menentukan tujuan sistem, dan aktor yang terlibat skenario *P2P lending* dari aplikasi yang di observasi.

3.2.2. *System Design*

Perancangan sistem ditulis menggunakan notasi UML, dengan menggunakan *Use Case Diagram, Activity Diagram* dan *Class Diagram*. Dalam tahap ini bisnis proses dari rancang bangun aplikasi ditulis, beserta dengan rancangan basis datanya.

3.2.3. *Coding And Testing*

Sesuai dengan namanya, pada tahap ini akan dilakukan pemrograman sistem yang dirancang, untuk menghasilkan aplikasi yang sesuai dengan rancangan pada tahap sebelumnya. Selain itu pula, dilakukan juga pengujian sistem yang selesai dibangun dengan menggunakan metode *BlackBox Testing*.

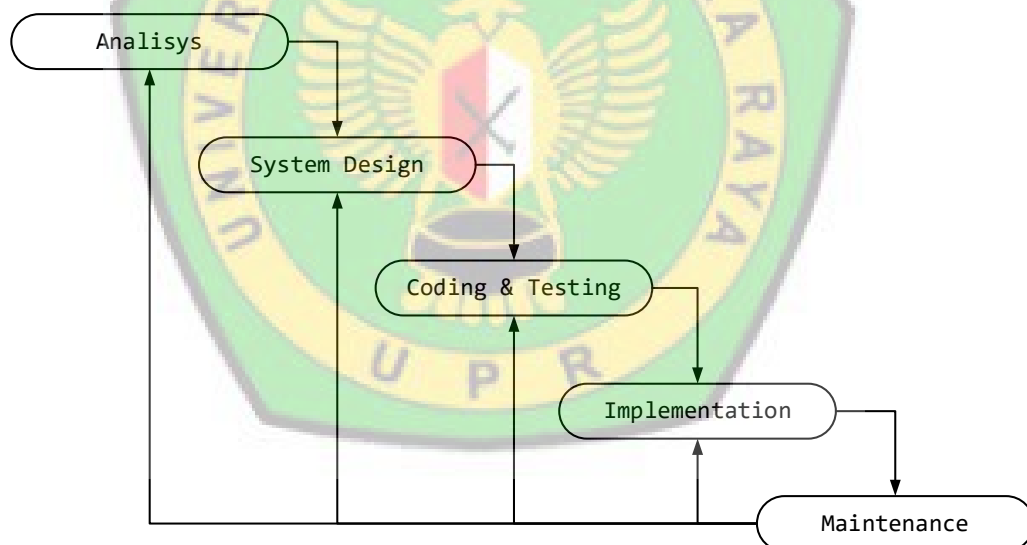
3.2.4. *Implementation*

Tahapan implementasi berbicara mengenai penerapan sebuah aplikasi yang selesai diuji kepada sebuah sistem berjalan dengan pengguna aktif dan transaksi yang nyata. Namun pada pembuatan rancang bangun aplikasi ini, implementasi akan dimaksud digantikan dengan simulasi transaksi yang dilakukan antara tiga aktor yang terkait dalam aplikasi, yaitu admin, pendana dan peminjam.

3.2.5. *Maintenance*

Aplikasi yang dibuat tidak membutuhkan pemeliharaan program, karena merupakan rancang bangun aplikasi, dan transaksi yang dilakukan merupakan sebuah simulasi transaksi.

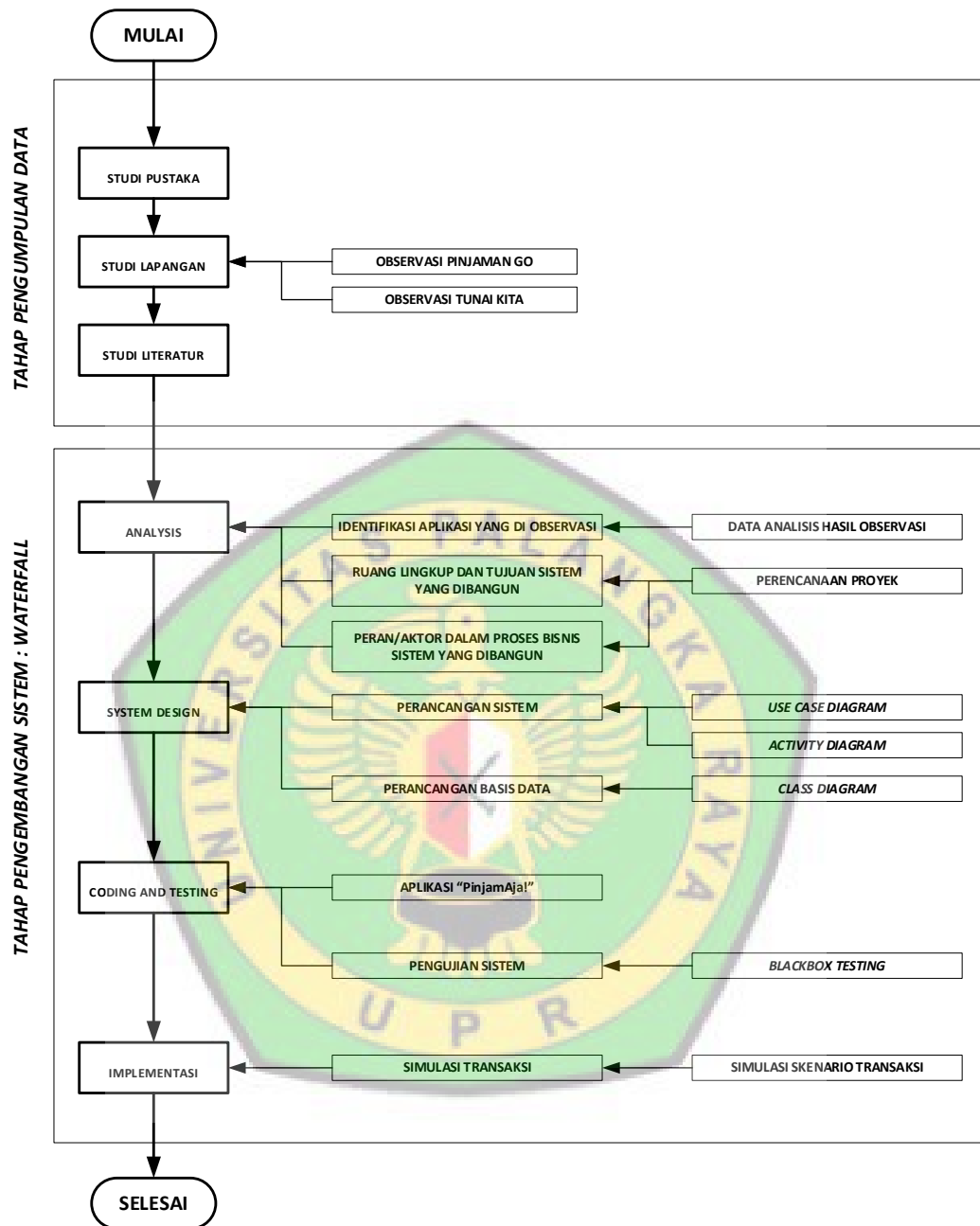
Pada gambar 3.1. diagram metode pengembangan sistem *waterfall* pada aplikasi PinjamAja.



GAMBAR 3.1. Metode Pengembangan Sistem *Waterfall* Pada Aplikasi “Pinjam Aja!”

3.3. KERANGKA PENELITIAN

Penyusunan hasil penelitian skripsi “Rancang Bangun Aplikasi Peminjaman Uang “Pinjam Aja!” Dengan Model P2P *Lending*” terdiri dari dua tahapan yaitu pertama dengan metode pengumpulan data yang terdiri dari studi lapangan dan studi literatur. Kedua, mengguna metode analisis dan perancangan sistem dengan menggunakan metode *Waterfall*.



GAMBAR 3.2. Kerangka Penelitian

3.4. PEUBAH DAN PENGUKURAN

TABEL 3.1. Peubah Dan Pengukuran

No.	Tahapan Penelitian	Indikator Capaian	Target
1.	Pengumpulan Data	Menemukan sumber Referensi, Menentukan aplikasi P2P sejenis yang di observasi, dan Teori pendukung.	Uraian Tinjauan Pustaka, Uraian teori pendukung pada Landasan Teori dan Bisnis proses dari aplikasi P2P yang di observasi.
2.	Analisis	Mengidentifikasi proses bisnis pada aplikasi yang di observasi, menentukan batasan ruang lingkup sistem yang dibangun dan menentukan tujuan sistem serta penentuan aktor/peran dalam proses bisnis aplikasi yang direncanakan.	Data analisis dari aplikasi yang di observasi, Perencanaan proyek.
3.	Desain Sistem	Perancangan sistem, Perancangan basis data dan Perancangan Antar Muka	Desain <i>Use Case Diagram</i> , <i>Activity Diagram</i> dan <i>Class Diagram</i>
4.	Pemrograman dan Pengujian	Aplikasi “Pinjam Aja!” dan Pengujian Sistem	Hasil tes pengujian <i>blackbox</i>
5.	Implementasi	Simulasi Transaksi	Simulasi skenario transaksi P2P antara admin, pendana dan peminjam.

3.5. DATA ANALISIS DARI APLIKASI YANG DI OBSERVASI

3.5.1. Observasi Bunga, Besaran Dan Syarat Pinjaman

A. TunaiKita

Berikut merupakan beberapa detail dari observasi yang didapat dari aplikasi Tunai Kita :

1. Batas Pinjaman adalah Rp. 700.000,- s.d. Rp. 2.100.000,- dengan waktu pelunasan pinjaman adalah 30 hari setelah pinjaman dikirim rekening pengguna.
2. Bunga pinjaman sebesar 24%.
3. Tidak ada fitur cicil bayar, ketika terhitung hari ke-30 pada akhir tenor, peminjam diminta untuk membayar jumlah pinjaman beserta bunga 24%.
4. Data utama yang diperlukan dalam pengajuan pinjaman antara lain foto KTP beserta informasi terkait sesuai dengan KTP, alamat tempat tinggal, pendidikan terakhir, nama ibu, kontak darurat, informasi pekerjaan, tujuan/alasan pinjaman dan No. rekening peminjam.
5. Besaran pinjaman dana yang dapat diajukan oleh pengguna dapat ditingkatkan dengan cara mengunggah data tambahan seperti NPWP, SIM A atau C, Slip Gaji/Rekening Koran, BPJS, Kartu Keluarga, Resi KTP, Tagihan PLN/PAM, Surat Kontrak Rumah/Apartemen dan Surat Domisili

B. PinjamanGo

Berikut merupakan beberapa detail dari observasi yang didapat dari aplikasi Pinjaman Go :

1. Besarnya dana pinjaman maksimal adalah Rp. 6.000.000,- dengan waktu lama pinjaman 3 bulan terhitung dari cairnya dana pinjaman dikirimkan ke pengguna. Jumlah pinjaman Rp.6.000.000,- adalah besaran dana yang harus di kembalikan peminjam. Besaran dana di transfer ke peminjam adalah sebesar Rp. 4.744.260,- setelah potongan rincian biaya tambahan termasuk biaya untuk pendana (termasuk bunga dan komisi partisipasi) sebesar Rp. 92.042,- , biaya layanan

platform sebesar Rp. 189.648,- , biaya layanan teknis Rp. 397.278,- , Premi asuransi Rp. 75.000,- dan Biaya Manajemen risiko Rp. 501.772,-

2. Ada tiga fase pengembalian pinjaman. Fase pertama 1-30 hari membayar sebesar Rp. 5.400.000,- (90% dari total pinjaman), fase kedua 31-60 hari membayar sebesar Rp. 300.000,- (5% dari total bayar) dan fase ketiga 61-90 hari membayar Sebesar 300.000,- (5% dari total bayar).
3. Bunga keterlambatan bayar 1.5% per hari.
4. Data yang diperlukan dalam pengajuan pinjaman antara lain Informasi Identitas, Informasi Pribadi, Informasi Pekerjaan dan kontak.

3.5.2. Hasil Observasi Alur Pengajuan Pinjaman

A. TunaiKita

1. Peminjam mendaftar dan melengkapi profil dengan menggunakan nomor ponsel pada platform TunaiKita
2. Peminjam dapat meningkatkan limit pinjaman dengan cara memasukkan data pribadi dengan lengkap untuk meningkatkan skor kredit.
3. Peminjam mengajukan pinjaman dan tunggu keputusan dalam waktu 12-24 hari kerja
4. Apabila Permintaan pinjaman diterima, peminjam menerima transfer di rekening bank

B. PinjamanGo

1. Peminjam mengajukan pinjaman di platform PinjamanGo
2. Tim PinjamanGo memproses pengajuan peminjam
3. Pendana memilih dan memberi pinjaman calon peminjam
4. Peminjam menerima pinjaman
5. Pendana menerima penghasilan bunga

3.5.3. Hasil Observasi Alur Pengajuan Pendanaan

A. TunaiKita

Tidak ada keterangan diberikan mengenai pendaftaran sebagai pendana.

B. PinjamanGo

1. Isi data pribadi pendana.
2. Data pendana di tinjau oleh admin.
3. Ketika permintaan diterima, pendana dapat memilih memberikan kepada peminjam melalui aplikasi
4. Penghasilan di terima berupa pinjaman pokok beserta bunga sebesar 16%

3.6. DESKRIPSI PROYEK

TABEL 3.2. Deskripsi Proyek

Nama Proyek	Rancang Bangun Aplikasi Peminjaman Uang “Pinjam Aja!” Dengan Model <i>P2P Lending</i>
Nama Aplikasi	<i>Pinjam Aja!</i> dan <i>Pinjam Aja! Admin</i>
Teknologi yang digunakan	Android, Java, NoSQL, <i>Firestorei</i> , <i>Cloud</i>
Jenis Aplikasi	Layanan Pinjam Meminjam Uang Berbasis Teknologi Informasi
Target Pengguna	Aplikasi ditujukan untuk masyarakat umum, berusia minimal 18 tahun

3.7. PERENCANAAN PROYEK

3.7.1. Bunga, Besaran Dan Syarat Pinjaman

1. Besar pinjaman dana yang dapat dipilih untuk di ajukan oleh peminjam melalui aplikasi Pinjam Aja! adalah sebesar Rp. 500.000,- , Rp. 1.000.000,- atau Rp. 1.500.000,-.
2. Bunga pinjaman sebesar 20%

3. Peminjam melunasi pengembalian pokok dan bunga pinjaman kredit dengan mode cicilan 3 kali, dengan pembayaran pertama dihitung pada bulan selanjutnya.
4. Bunga pinjaman keterlambatan pembayaran adalah 0,8% per hari.
5. Data penting untuk pengajuan pinjaman antara lain, foto KTP, alamat tempat tinggal, pendidikan terakhir, pekerjaan, nama ibu, kontak orang terdekat dan nomor rekening.
6. Data penting untuk pengajuan pendana antara lain, foto KTP, alamat tempat tinggal, pendidikan terakhir, pekerjaan, nama ibu, kontak orang terdekat dan nomor rekening.
7. Penetapan jumlah total biaya, biaya keterlambatan dan seluruh biaya-biaya lainnya sebesar maksimal 100% (seratus persen) dari nilai pinjaman.

3.7.2. Peran/Aktor Dalam Proses Bisnis

1. Admin

Peran : mengatur jalannya proses administrasi dari aplikasi, termasuk di dalamnya mengelola pendaftaran peminjam dan pendana serta mengirim dan menerima transfer tunai dalam transaksi antara peminjam dan pendana.

2. Pendana

Peran : pemberi dana pinjaman kepada peminjam, melihat informasi mengenai permintaan pinjaman dan dapat memilih peminjam mana yang didanai.

3. Peminjam

Peran : Mengajukan pinjaman melalui aplikasi dan melunasi pengembalian pokok pinjaman beserta bunga pinjaman di akhir masa tenor.

3.7.3. Proses Bisnis Alur Pengajuan Pinjaman

1. Peminjam mendaftar melalui aplikasi dan memasukkan data penting untuk permintaan pengajuan pinjaman.

2. Peminjam dapat melihat opsi pengajuan pinjaman, tapi sebelum dapat melakukan permintaan pinjaman, peminjam diharuskan melengkapi diminta untuk melengkapi data pendaftaran.
3. Ketika data pendaftaran sudah lengkap, peminjam dapat memilih mengajukan permintaan pinjaman.
4. Ketika pengajuan pinjaman diverifikasi oleh admin, barulah permintaan pengajuan pinjaman dapat dilihat pada *dashboard* pendana.
5. Saat Pendana memilih untuk mendanai pinjaman, pendana mentransfer tunai ke rekening *escrow* senilai dengan besaran permintaan pinjaman.
6. Admin memverifikasi transfer dari pendana, dan memperbaharui data terkait pendanaan dalam akun virtual pendana dan akun virtual peminjam.
7. Peminjam menerima transfer pencairan dana pinjaman.
8. Pada hari akhir masa tenor, peminjam diwajibkan sudah melunasi pokok pinjaman beserta bunga sebesar 20%.

3.7.4. Proses Bisnis Alur Pengajuan Pendanaan

1. Pendana mendaftar melalui aplikasi dan memasukkan data penting yang di perlukan.
2. Pendana memiliki akses untuk melihat data permintaan pinjaman dan dapat memilih pinjaman mana yang didanai.
3. Sebelum pendana dapat melakukan pendanaan, pendana diminta untuk melengkapi data pendaftaran pendana.
4. Ketika data pendaftaran sudah lengkap, pendana dapat memilih permintaan mana yang akan didanai.
5. Data pengajuan pendanaan ditinjau oleh admin, admin menunggu transfer ke rekening *escrow*.
6. Di akhir masa tenor pinjaman yang didanai, pendana menerima pengembalian pokok pinjaman dan 15% bunga pinjaman.

3.7.5. Kalkulasi Simulasi Pinjaman

1. Besar Pinjaman : Rp. 1.500.000,-

Waktu Pinjaman : 90 Hari

Mode Cicilan : Bayar Per bulan

Bunga 20% : Rp. 300.000,-

Total Pembayaran : Rp. 1.800.000,-

Besar Cicilan : 3 x Rp. 600.000,-

Denda : Total Pembayaran x 0,8% per hari = Rp. 14.400,- per hari

Bunga + Denda Maksimal : 100% dari Besar Pinjaman + Bunga 20%

Maksimal Dana Kembali : Total Pembayaran + Denda = Rp. 3.600.000,-

2. Besar Pinjaman : Rp. 1.000.000,-

Waktu Pinjaman : 90 Hari

Mode Cicilan : Bayar Per bulan

Bunga 20% : Rp. 200.000,-

Total Pembayaran : Rp. 1.200.000,-

Besar Cicilan : 3 x Rp. 400.000,-

Denda : Total Pembayaran x 0,8% per hari = Rp. 9.600,- per hari

Bunga + Denda Maksimal : 100% dari Besar Pinjaman + Bunga 20%

Maksimal Dana Kembali : Total Pembayaran + Denda = Rp. 2.400.000,-

3. Besar Pinjaman : Rp. 500.000,-

Waktu Pinjaman : 90 Hari

Mode Cicilan : Bayar Per bulan

Bunga 20% : Rp. 100.000,-

Total Pembayaran : Rp. 600.000,-

Besar Cicilan : 3 x Rp. 200.000,-

Denda : Total Pembayaran x 0,8% per hari = Rp. 4.800,- per hari

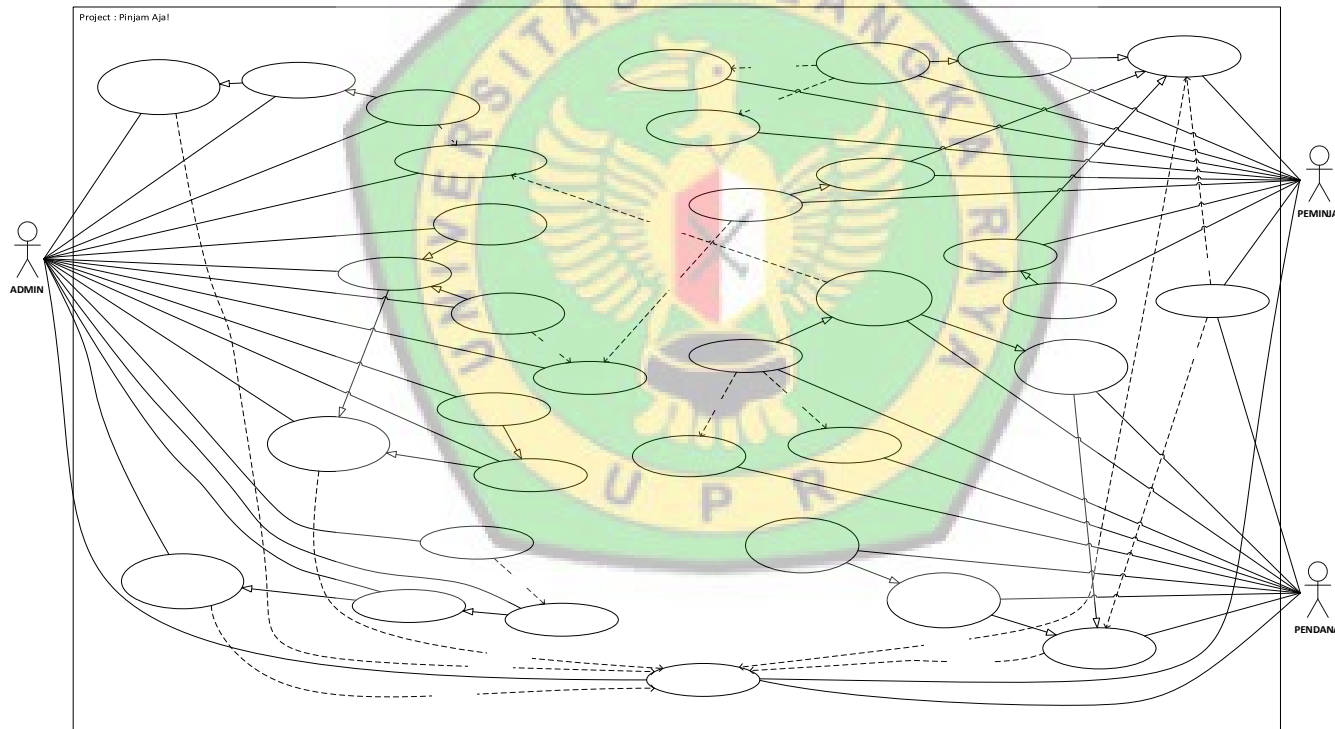
Bunga + Denda Maksimal : 100% dari Besar Pinjaman + Bunga 20%

Maksimal Dana Kembali : Total Pembayaran + Denda = Rp. 1.200.000,-

3.8. DESAIN SISTEM

3.8.1. Usecase Diagram

Berikut pada gambar 3.3. merupakan diagram *Use Case* dari perencanaan proyek Pinjam Aja!, dengan tiga aktor/peran dalam aplikasi, yaitu Admin, Pendana dan Peminjam. Terdapat dua aplikasi yang dikembangkan, yang saling terhubung dalam satu basis data di *Firebase*, aplikasi Pinjam Aja! untuk aktor peminjam dan aktor pendana, serta aplikasi Pinjam Aja! Admin untuk aktor admin.

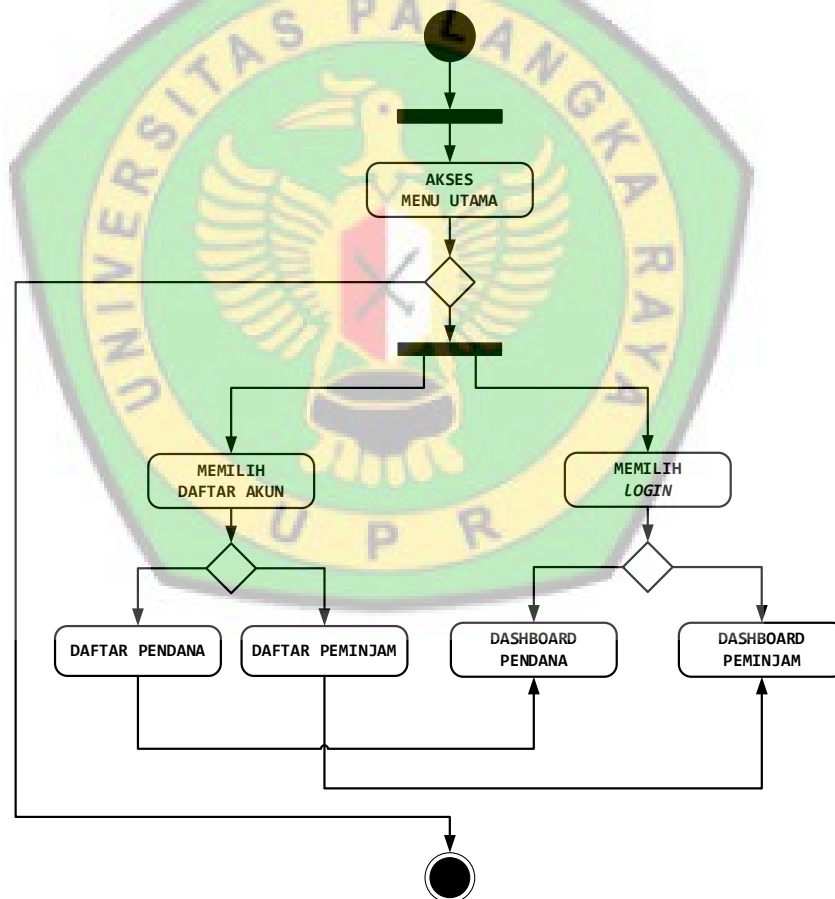


GAMBAR 3.3. Use Case Diagram

3.8.2. Activity Diagram

Pada gambar 3.4. adalah diagram aktivitas ketika aplikasi Pinjam Aja! (untuk sisi *user* pendana dan peminjam) dijalankan. Terdapat opsi untuk *login* dan untuk mendaftar akun.

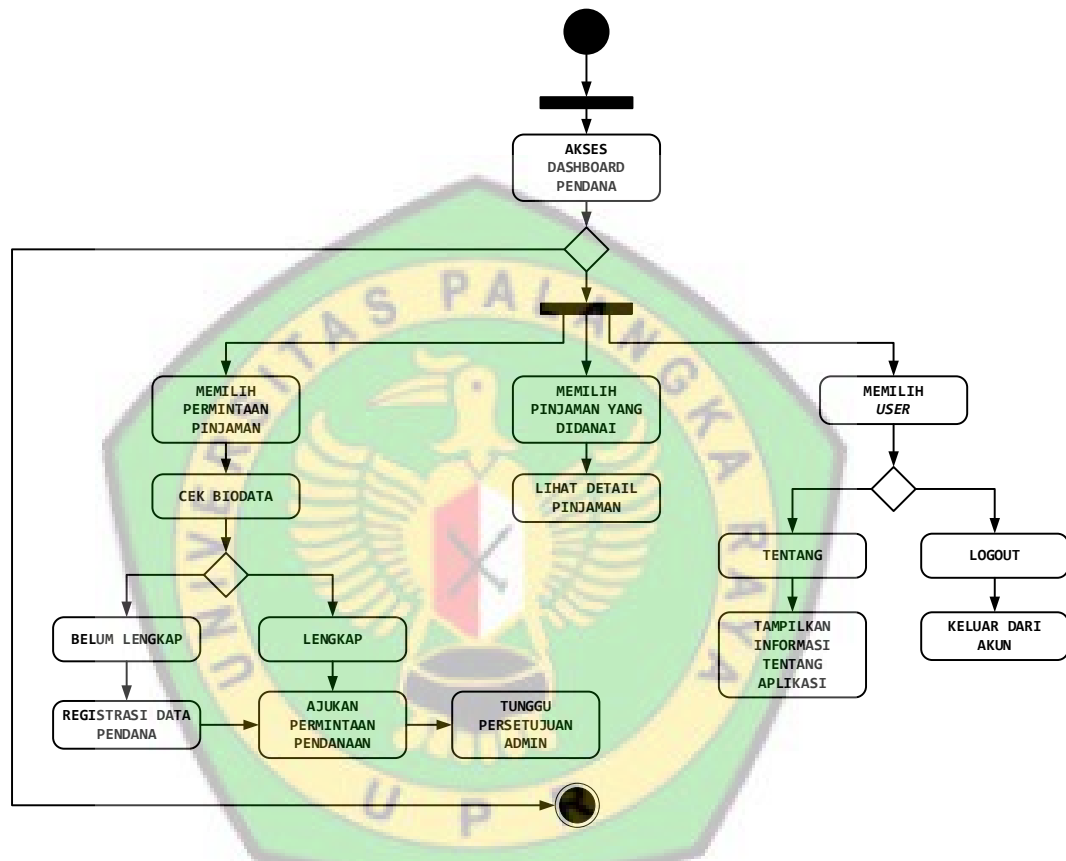
Ketika aktor memilih daftar akun, aktor akan di arahkan ke *form* pendaftaran dengan opsi untuk dapat memilih jenis akun yang di daftar, apakah sebagai peminjam atau pendana. Jika aktor memilih *login*, user akan di arahkan ke *dashboard* yang sesuai dengan jenis akun yang telah di daftarkan, apakah sebagai pendana atau peminjam. Untuk akun admin yang mencoba *login* di sisi pengguna, aplikasi akan menampilkan permintaan *login* ditolak.



GAMBAR 3.4. Activity Diagram Case Daftar/Login Pinjam Aja! Untuk User Peminjam Dan Pendana

Gambar 3.5. adalah diagram aktivitas untuk aktor peminjam. Terdapat pilihan untuk memilih permintaan pinjaman dari opsi yang tersedia, yaitu pinjaman Rp.

Pada Gambar 3.6. adalah diagram aktivitas untuk aktor pendana, pada opsi permintaan pinjaman, pendana dapat melihat daftar permintaan pendanaan yang telah disetujui oleh admin. Aktor pendana dapat memilih pinjaman mana yang didanai. Untuk permintaan pendanaan pertama kali pendana diminta melengkapi biodata pendana, baru kemudian mengajukan permintaan pendanaan.

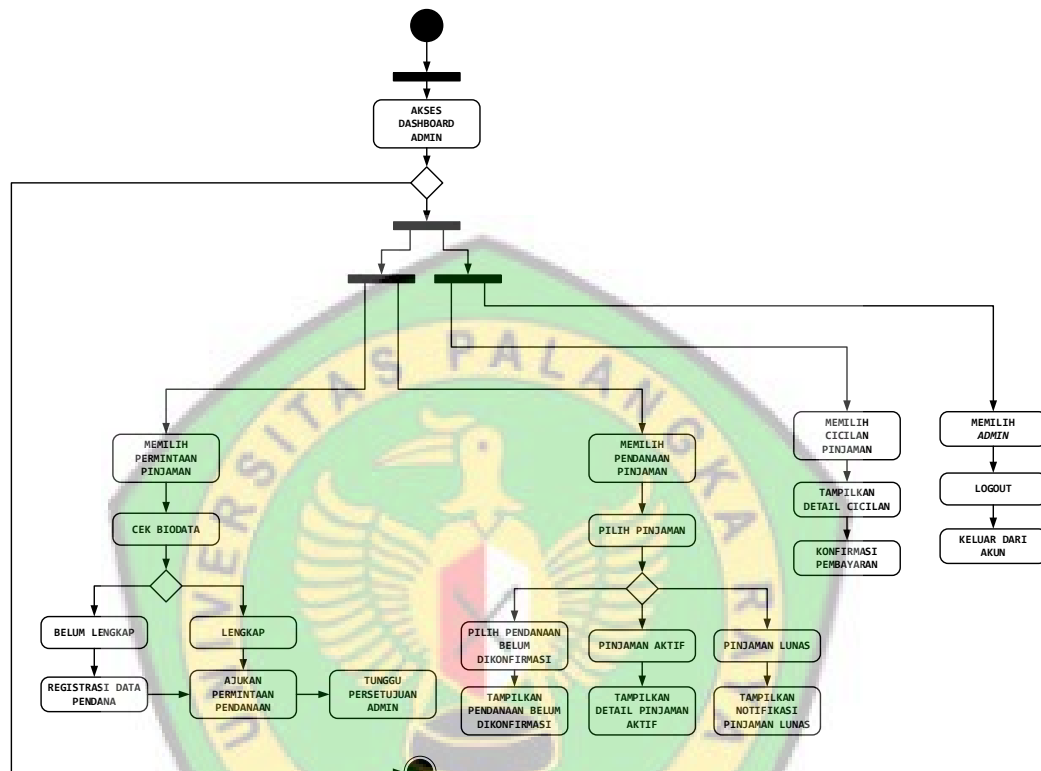


GAMBAR 3.6. Activity Diagram Case Dashboard Pendana

Pada gambar 3.7. adalah diagram aktivitas untuk aktor *admin* dalam aplikasi *Pinjam Aja! Admin*. Dalam opsi permintaan pinjaman aktor admin dapat melihat daftar permintaan pinjaman dan melakukan *review* terhadap data peminjam, apabila biodata pendana sudah lengkap, admin dapat menyetujui permintaan pinjaman untuk dapat tampil dalam daftar permintaan pendanaan pada *dashboard* pendana.

Pada opsi pendanaan, admin dapat melihat seluruh daftar permintaan pinjaman yang didanai oleh pendana, termasuk di dalamnya permintaan pendanaan pinjaman yang membutuhkan konfirmasi admin, pinjaman aktif dan pinjaman yang sudah lunas.

Pada opsi Cicilan, admin dapat melihat daftar pembayaran cicilan yang dilakukan oleh peminjam, dan mengkonfirmasi apabila transfer cicilan sudah diterima ke rekening *escrow* dan pengembalian dana beserta bunga 15% sudah di transfer kembali ke pendana.

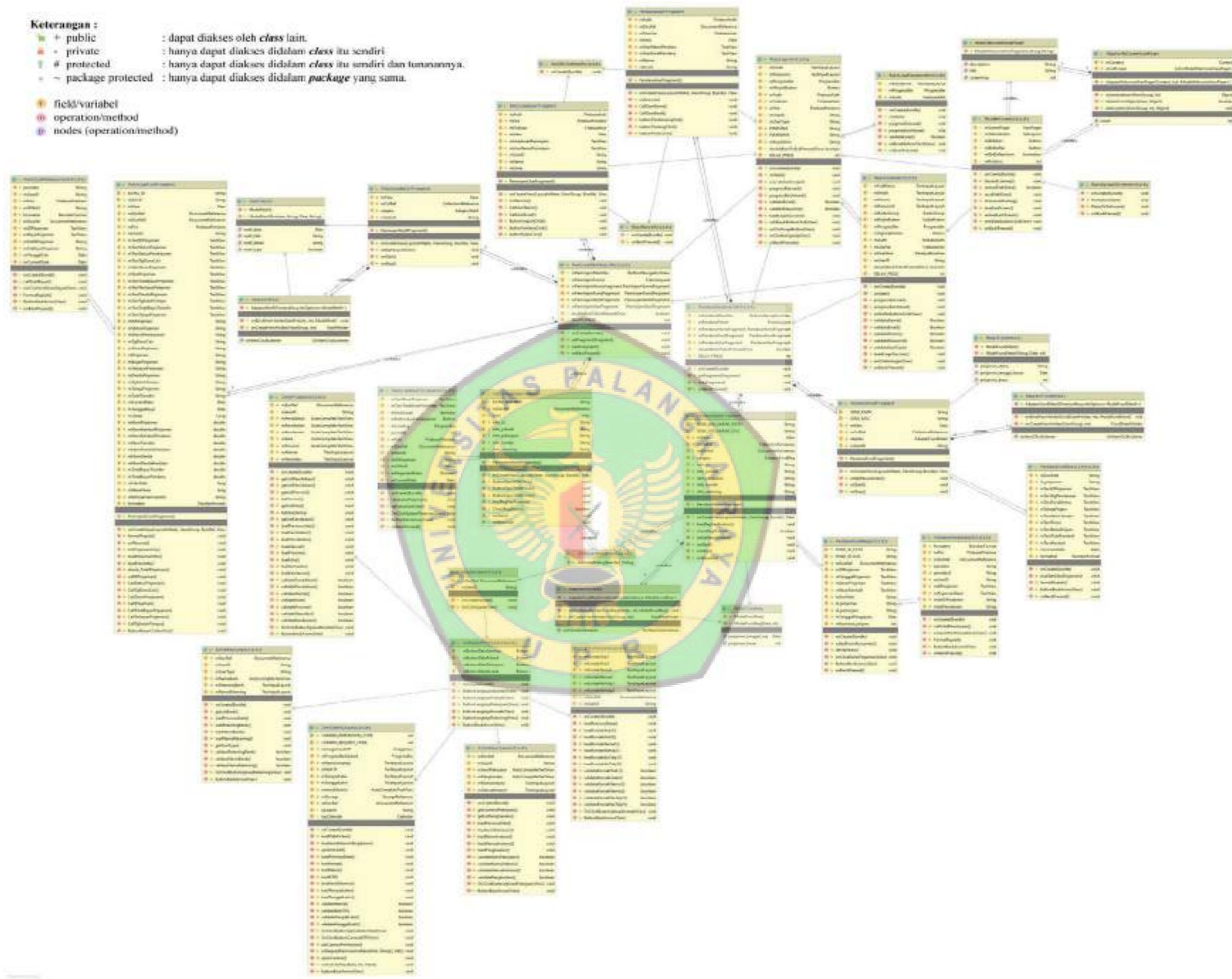


GAMBAR 3.7. Activity Diagram Case Dashboard Admin

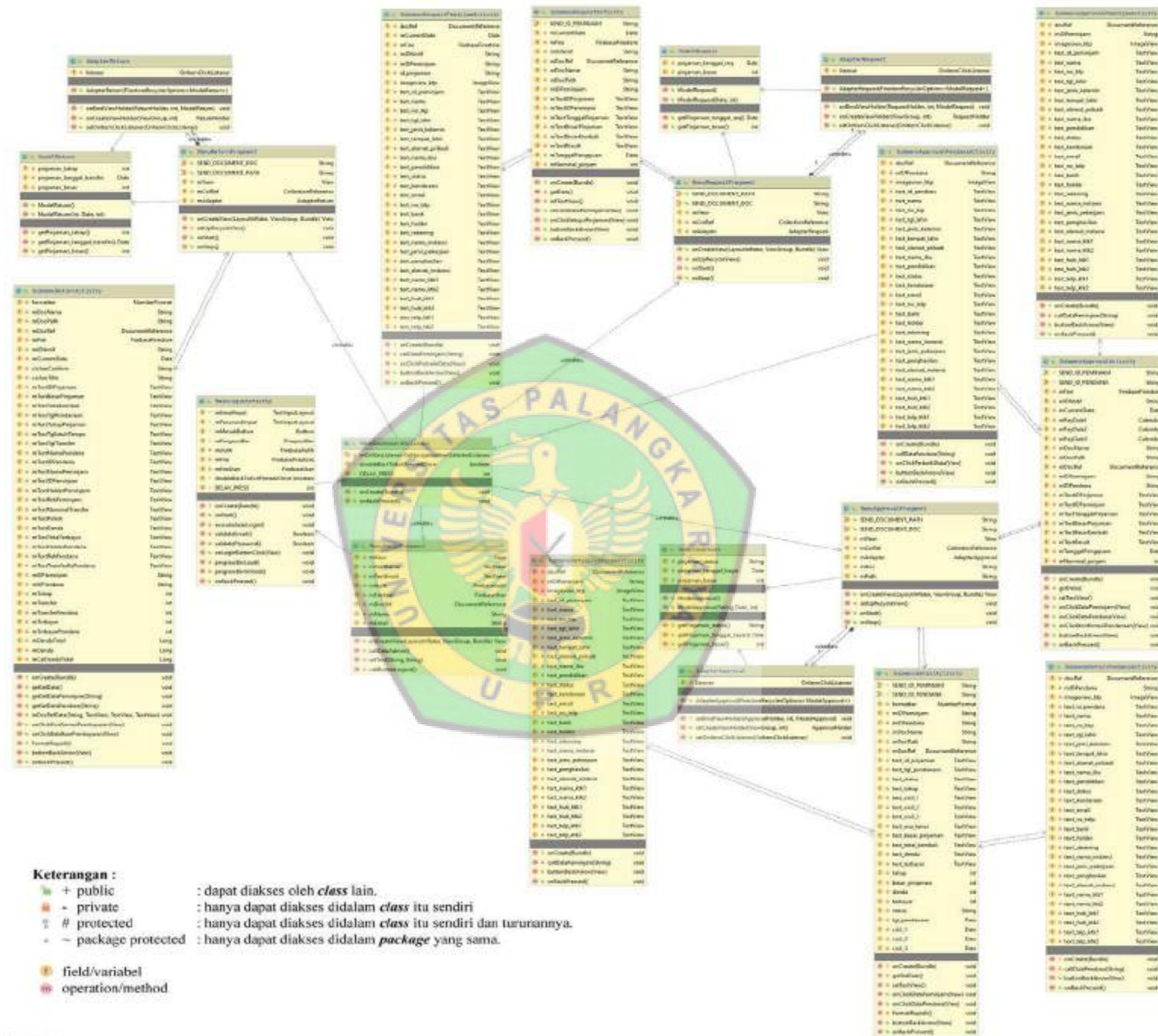
3.8.3. Class Diagram

Berikut merupakan masing-masing *class diagram* dari kedua aplikasi yang dikembangkan, pada gambar 3.8. merupakan *class diagram* dari aplikasi PinjamAja! Sisi pengguna (peminjam dan pendana). Berikutnya pada gambar 3.9. merupakan *class diagram* dari aplikasi PinjamAja! Admin, untuk sisi admin dari sistem.

Aplikasi PinjamAja! untuk sisi pengguna, digunakan peminjam untuk mengajukan pinjaman, mengecek pinjaman aktif serta mengecek tanggal pembayaran cicilan. Aplikasi PinjamAja! Admin, digunakan admin untuk memverifikasi biodata pendana dan peminjam, serta memverifikasi permintaan pinjaman dan pendanaan.



GAMBAR38. Class Diagram Dari Aplikasi PinjamAja!



GAMBAR39. Class Diagram Dari Aplikasi PinjamAja! Admin

BAB IV HASIL DAN PEMBAHASAN

4.1. KEBUTUHAN SISTEM

Berikut merupakan detail kebutuhan perangkat keras dan kebutuhan perangkat lunak untuk pengembangan proyek “Pinjam Aja!”.

A. Perangkat Keras (*Hardware*)

a. Laptop dengan spesifikasi sebagai berikut :

- 1) CPU : AMD *Dual Core* A9-9420 @ 3GHz
- 2) VGA : AMD Radeon R5
- 3) RAM : 8 GB
- 4) HDD : 500 GB
- 5) LCD : 15.6” – 1366x768 pixels
- 6) OS : Windows 10 Enterprise

b. *Smartphone Android* untuk *testing* dengan spesifikasi sebagai berikut :

- 1) CPU : Octa-core 1.40 GHz Cortex-A53
- 2) CHIPSET : Qualcomm MSM8940 Snapdragon 435
- 3) GPU : Adreno 505
- 4) RAM : 2 GB
- 5) STORAGE : *Internal* : 16 GB
- 6) LCD : IPS LCD 16M Colors 5” 720x1280 pixels
- 7) OS : MIUI 9.1.1.0 (NAMMIEI) *based on Android 7.1.2*

B. Perangkat Lunak (*Software*)

- a. Android Studio 3.6.0: Sebagai IDE (*Integrated Development Enviroment*).
- b. Adobe Photoshop CC 2019: Digunakan untuk membuat komponen visual dalam aplikasi.
- c. Adobe Illustrator CC 2019: Digunakan untuk membuat komponen visual dalam aplikasi.
- d. Adobe XD : Digunakan untuk perancangan *layout* aplikasi.

4.2. IMPLEMENTASI DAN EVALUASI SISTEM

Pada bagian implementasi sistem, hasilnya akan menunjukkan bagaimana desain sistem dari sebuah aplikasi diterapkan. Sedangkan pada evaluasi sistem, akan menampilkan hasil evaluasi setiap fungsi aplikasi menggunakan *blackbox testing*.

4.2.1. Implementasi

Proyek Pinjam Aja! terpisah menjadi dua bagian aplikasi berdasarkan jenis Aktornya. Aplikasi pertama dari sisi *user*, untuk mengakomodasikan transaksi permintaan pinjaman dan pendanaan. Aplikasi kedua dari sisi *admin* untuk mengevaluasi data pendana dan peminjam, mengelola permintaan pinjaman, mengkonfirmasi permintaan pendanaan dan mengkonfirmasi pembayaran cicilan.

4.2.1.1. Tampilan Aplikasi Sisi Pengguna

1. *Activity Splash Screen*

Tampilan *activity* pada gambar 4.1. merupakan *splash screen* dari aplikasi Pinjam Aja! yang ditampilkan ketika aplikasi mulai dijalankan.



GAMBAR 4.1. *Splash Screen* aplikasi *Pinjam Aja!*

2. *Activity Selamat Datang*

Tampilan *activity* pada gambar 4.2. merupakan fitur selamat datang dari aplikasi, yang hanya terbuka ketika aplikasi pertama kali di *install* dan dijalankan oleh pengguna.



GAMBAR 4.2. Activity Selamat Datang

3. Activity Daftar Pengguna (Pendana dan Peminjam)

Tampilan *activity* pada gambar 4.3. merupakan tampilan untuk mendaftar akun pada aplikasi Pinjam Aja!. Detail profil pengguna lebih lanjut akan diminta ketika pinjaman mengakses permintaan pinjaman untuk pertama kali, dan untuk pendana, detail profil diminta ketika pendana melakukan pendanaan pertama kali.

GAMBAR 4.3. Daftar Pengguna

4. Activity Login Pengguna

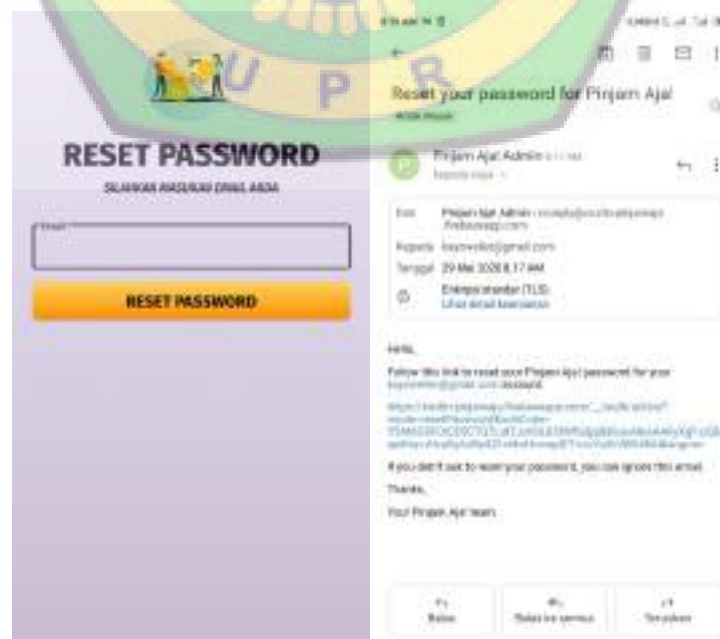
Tampilan *activity* pada gambar 4.4. berikut merupakan tampilan untuk melakukan login ke dalam aplikasi.



GAMBAR 4.4. Login Pengguna

5. Activity Lupa Password

Tampilan *activity* pada gambar 4.5. merupakan tampilan untuk melakukan permintaan lupa sandi, dengan menuliskan email pengguna, penerima akan menerima sebuah email berupa *link* untuk melakukan reset kata sandi.



GAMBAR 4.5. Lupa Password

6. *Activity Dashboard* Peminjam : Beranda

Berikut pada gambar 4.6. merupakan tampilan pada bagian *dashboard* peminjam pada opsi beranda, disini peminjam dapat melakukan permintaan pinjaman. Ketika peminjam melakukan permintaan pinjaman pertama kali, peminjam diminta untuk melengkapi profilnya sebelum dapat melakukan permintaan pinjaman.



GAMBAR 4.6. *Dashboard* Peminjam opsi Beranda

Pada Gambar 4.7. merupakan dialog yang ditampilkan ketika biodata untuk profil peminjam belum dilengkapi.



GAMBAR 4.7. Dialog permintaan untuk melengkapi biodata pengguna

Pada gambar 4.8. merupakan tampilan aktivitas untuk *menginput* biodata lengkap peminjam, proses ini hanya dilakukan sekali saja ketika peminjam melakukan permintaan pinjaman pertama kali.



GAMBAR 4.8. Pengisian Biodata Peminjam

Pada gambar 4.9. merupakan tampilan untuk melakukan permintaan pinjaman, ketika permintaan pinjaman dilakukan, admin akan memverifikasi terlebih dahulu permintaan pinjaman sebelum ditampilkan pada *dashboard* permintaan pinjaman pada pendana agar siap didanai.



GAMBAR 4.9. Permintaan Pinjaman

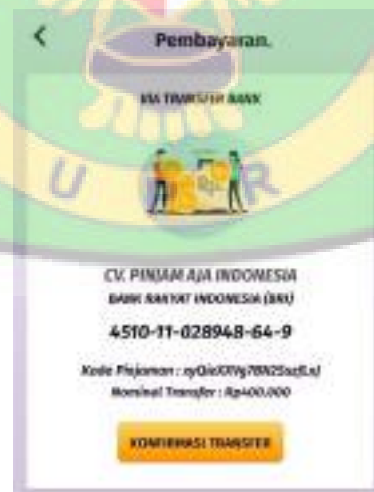
7. Activity Dashboard Peminjam : Dana

Pada bagian ini, seperti tampak pada gambar 4.10. ditampilkan keterangan mengenai pinjaman yang dilakukan, seperti status pinjaman, tanggal pencairan pinjaman, sisa waktu pinjaman, paket pinjaman, total bayar dan cicilan terbayar serta denda.



GAMBAR 4.10. Keterangan Pinjaman yang Aktif

Selain itu juga terdapat tombol untuk melakukan pembayaran cicilan, yang hanya bisa digunakan ketika ada pinjaman aktif. Jika peminjam melakukan pembayaran, admin akan mengecek apakah pembayaran cicilan sudah masuk ke dalam rekening *escrow*, jika sudah admin akan memverifikasi transfer. Gambar 4.11. menunjukkan tampilan aktivitas transfer cicilan.



GAMBAR 4.11. Transfer Cicilan

8. *Activity Dashboard* Peminjam : Notifikasi

Pada bagian, seperti tampak pada gambar 4.12. program menampilkan notifikasi dari proses transaksi yang terjadi dalam aplikasi, diantaranya,

notifikasi pengajuan pinjaman, notifikasi pendanaan, dan notifikasi pembayaran cicilan.



GAMBAR 2.12 Dashboard : Notifikasi

9. Activity Dashboard Peminjam : Profil

Pada bagian ini, seperti tampak pada gambar 4.13. ditampilkan nama dan email dari pengguna, selain itu ada tombol tentang untuk menampilkan tentang aplikasi dan tombol *logout* untuk keluar dari akun pengguna.



GAMBAR 4.13. Dashboard : Profil Peminjam

10. Activity Dashboard Pendana : Beranda

Pada bagian opsi *dashboard* ini, seperti tampak pada gambar 4.14, pendana dapat melihat daftar permintaan pinjaman yang siap didanai, dimana profil peminjam sudah di verifikasi oleh admin, informasi pinjaman yang dapat dilihat oleh pendana, hanya merupakan informasi yang di izinkan menurut aturan OJK, oleh karena itu pendana tidak di izinkan untuk dapat mengakses profil peminjam.



GAMBAR 4.14. Permintaan pendanaan pada beranda Pendana

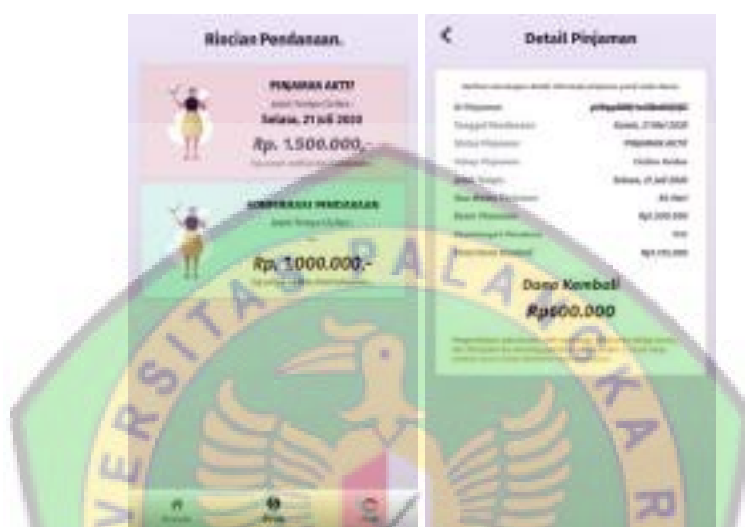
Sama seperti pada *dashboard* peminjam, peminjam diminta untuk melengkapi biodatanya terlebih dahulu untuk dapat melakukan pendanaan. Ketika data pendana sudah lengkap, barulah pendana dapat memilih permintaan mana yang di danai kemudian mentransfer pendanaan sesuai dengan besaran pinjaman yang diminta.



GAMBAR 4.15. Pendanaan Pinjaman

11. Activity Dashboard Pendana : Dana

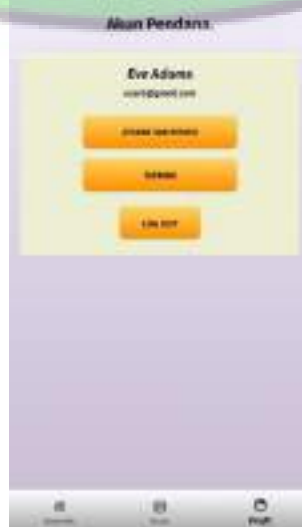
Seperti terlihat pada gambar 4.16., *dashboard fragment* dana menampilkan daftar mengenai pinjaman yang didanai oleh pendana. Disini pendana dapat melihat sejauh mana *progress* dari pinjaman yang berjalan, dan informasi penting seperti kapan jatuh tempo dari pinjaman, *progress* cicilan dan berapa besar cicilan yang sudah dibayar.



GAMBAR 4.16. Fragment Dana dan Submenu detail Pinjaman

12. Activity Dashboard Pendana : Profil

Pada bagian ini, seperti tampak pada gambar 4.17. ditampilkan nama dan email dari pendana, selain itu ada tombol tentang untuk menampilkan tentang aplikasi dan tombol *logout* untuk keluar dari akun pengguna.



GAMBAR 4.17. Dashboard User : Profil Pendana

4.2.1.2. Tampilan Aplikasi Sisi Admin

1. Activity Login

Sebelum admin dapat mengelola data peminjam dan pendana, serta melakukan verifikasi transaksi di antara keduanya, admin diminta untuk *login* terlebih dahulu. Berikut pada gambar 4.18 merupakan tampilan *login* dari admin.



GAMBAR 4.18. Activity Login Admin

2. Fragment Permintaan Pinjaman

Daftar setiap permintaan pinjaman yang dilakukan peminjam akan ditampilkan pada *fragment* ini, admin bertugas untuk memverifikasi permintaan pinjaman, termasuk didalamnya memverifikasi biodata peminjam apakah valid dapat melakukan pinjaman. Berikut merupakan tangkapan layar pada *fragment* ini dan sub menu terkait didalamnya.



GAMBAR 4.18. Fragment Permintaan Pinjaman dan Submenu

4. *Fragment* Cicilan

Daftar pembayaran cicilan ditampilkan pada *fragment* ini untuk diverifikasi oleh admin, verifikasi yang dilakukan yaitu mengecek apakah transfer sudah diterima ke rekening *escrow* dan dana sudah di transfer kembali ke rekening pendana beserta bunga 15%. Berikut merupakan tampilan pada *fragment* cicilan dan submenu untuk melakukan verifikasi transfer.



GAMBAR 4.21. *Fragment* Cicilan beserta Submenu

5. *Fragment* Admin

Fragment ini merupakan pilihan terakhir pada *dashboard* admin, terdapat satu tombol yang berfungsi untuk *logout* dari akun admin. Berikut pada gambar 4.22. merupakan tampilan *fragment* Admin.



GAMBAR 4.22. *Fragment* Admin

4.2.2. Evaluasi Sistem

Setelah implementasi diselesaikan, tahap selanjutnya adalah melakukan pengujian sistem apakah sudah berjalan sesuai dengan hasil yang diharapkan dari hasil perencanaan. Pengujian untuk melihat apakah ada kesalahan dan kekurangan pada sistem menggunakan metode *black box*.

Metode pengujian *blackbox*, fokus kepada kebutuhan fungsionalitas dari aplikasi, dengan melakukan pengujian apakah proses terhadap perintah yang dilakukan sesuai dengan yang diharapkan atau sebaliknya mengalami kendala.

4.2.2.1. Aplikasi Sisi Pengguna

Berikut merupakan parameter dan hasil pengujian *blackbox* yang dilakukan dari aplikasi pada sisi pengguna yaitu untuk peminjam dan pendana.

TABEL 4.1. Pengujian *Blackbox*, Daftar User

No.	Aksi	Hasil	Keterangan
1	Tap Daftar : Input tidak sesuai, atau ada kolom kosong	Tampilkan Letak Kesalahan	✓
2	Tap Daftar : Setiap kolom pendaftaran terisi dengan benar	Daftar Sukses	✓

TABEL 4.2. Pengujian *Blackbox*, Login User

No.	Aksi	Hasil	Keterangan
1	Tap <i>Login</i> : <i>Input</i> tidak sesuai, atau ada kolom kosong	Tampilkan Letak Kesalahan	✓
2	Tap <i>Login</i> : Email dan <i>password</i> terisi dengan benar	Login Sukses	✓

TABEL 4.3. Pengujian *Blackbox*, Lupa *password*

No.	Aksi	Hasil	Keterangan
1	Tap <i>Reset Password</i> : Format penulisan email salah	Tampilkan Pesan Kesalahan	✓
2	Tap <i>Reset Password</i> : Email terdaftar pada basis data	Kirim Email Reset Password Sukses	✓

TABEL 4.3. Pengujian *Blackbox*, Lupa password (Lanjutan)

No.	Aksi	Hasil	Keterangan
3	Tap <i>Reset Password</i> : Email tidak terdaftar	Tampilkan pesan kesalahan	✓

TABEL 4.4. Pengujian *Blackbox*, Dashboard Peminjam : *Fragment Beranda*

No.	Aksi	Hasil	Keterangan
1	Tap Permintaan Pinjaman: Biodata Peminjam belum lengkap	Tampilkan dialog untuk melengkapi data peminjam	✓
2	Tap Permintaan Pinjaman: Biodata Peminjam lengkap	Tampilkan Rincian Opsi Pinjaman	✓
3	Ajukan Permintaan Pinjaman	Permintaan Pinjaman Sukses	✓

TABEL 4.5. Pengujian *Blackbox*, Pengisian Biodata Peminjam

No.	Aksi	Hasil	Keterangan
1	Tap Informasi Identitas	Buka <i>Activity</i> Biodata Identitas	✓
2	Tap Informasi Pribadi	Buka <i>Activity</i> Biodata Pribadi	✓
3	Tap Informasi Pekerjaan	Buka <i>Activity</i> Biodata Pekerjaan	✓
4	Tap Informasi Kontak	Buka <i>Activity</i> Biodata Kontak	✓
5	Tap Informasi Rekening	Buka <i>Activity</i> Biodaya Rekening	✓

TABEL 4.5. Pengujian *Blackbox*, Pengisian Biodata Peminjam (*Lanjutan*)

No.	Aksi	Hasil	Keterangan
6	Buka Kamera untuk foto KTP	Kamera berhasil dibuka	✓
7	Tap Upload data, kondisi kolom terisi tidak lengkap	Upload data gagal	✓
8	Tap Upload data, kondisi kolom terisi lengkap	Upload data berhasil	✓

**TABEL 4.6. Pengujian *Blackbox*, Dashboard Peminjam :
*Fragment Rincian Pinjaman***

No.	Aksi	Hasil	Keterangan
1	Rincian Pinjaman	Ditampilkan sesuai	✓
2	Tap Bayar Pinjaman: Belum ada pinjaman aktif	Hanya Bisa diakses ketika ada pinjaman aktif	✓
3	Tap Bayar Pinjaman: Ada pinjaman aktif	Tampilkan Pembayaran Pinjaman	✓
4	Tap Konfirmasi Transfer	Permintaan konfirmasi transfer diterima admin	✓

**TABEL 4.7. Pengujian *Blackbox*, Dashboard Peminjam :
*Fragment Profil***

No.	Aksi	Hasil	Keterangan
1	Tap Tentang	Tampilkan Tentang Aplikasi	✓
2	Tap Logout	Keluar dari akun peminjam yang sedang aktif	✓

**TABEL 4.8. Pengujian *Blackbox*, Dashboard Pendana :
Fragment Beranda**

No.	Aksi	Hasil	Keterangan
1	Tap Permintaan Pinjaman: Biodata Pendana belum lengkap	Tampilkan dialog untuk melengkapi data peminjam	✓
2	Tap Permintaan Pinjaman: Biodata Pendana lengkap	Tampilkan Rincian Pengajuan Pinjaman	✓
3	Tap Danai Permintaan Pinjaman	Buka Activity Transfer Pendanaan	✓
4	Tap Konfirmasi Transfer	Konfirmasi Pendanaan dan transfer Menunggu verifikasi Admin	✓

TABEL 4.9. Pengujian *Blackbox*, Pengisian Biodata Peminjam

No.	Aksi	Hasil	Keterangan
1	Tap Informasi Identitas	Buka <i>Activity</i> Biodata Identitas	✓
2	Tap Informasi Pribadi	Buka <i>Activity</i> Biodata Pribadi	✓
3	Tap Informasi Pekerjaan	Buka <i>Activity</i> Biodata Pekerjaan	✓
4	Tap Informasi Kontak	Buka <i>Activity</i> Biodata Kontak	✓
5	Tap Informasi Rekening	Buka <i>Activity</i> Biodaya Rekening	✓
6	Buka Kamera untuk foto KTP	Kamera berhasil dibuka	✓

TABEL 4.9. Pengujian *Blackbox*, Pengisian Biodata Pendana (Lanjutan)

No.	Aksi	Hasil	Keterangan
7	Tap Upload data, kondisi kolom terisi tidak lengkap	Upload data gagal	✓
8	Tap Upload data, kondisi kolom terisi lengkap	Upload data berhasil	✓

**TABEL 4.10. Pengujian *Blackbox*, Dashboard Pendana :
Fragment Rincian Pendanaan**

No.	Aksi	Hasil	Keterangan
1	Tap Pinjaman	Buka Rincian Pinjaman yang didanai	✓

**TABEL 4.11. Pengujian *Blackbox*, Dashboard Pendana :
Fragment Profil**

No.	Aksi	Hasil	Keterangan
1	Tap Tentang	Tampilkan Tentang Aplikasi	✓
2	Tap Logout	Keluar dari akun pendana yang sedang aktif	✓

4.2.2.2. Aplikasi Sisi Admin

Berikut merupakan serangkaian parameter uji *blackbox testing* pada aplikasi dari sisi admin.

TABEL 4.12. Pengujian *Blackbox*, Login Admin

No.	Aksi	Hasil	Keterangan
1	Tap <i>Login</i> : <i>Input</i> tidak sesuai, atau ada kolom kosong	Tampilkan Letak Kesalahan	✓
2	Tap <i>Login</i> : Email dan <i>password</i> terisi dengan benar	Login Sukses	✓

**TABEL 4.13. Pengujian Blackbox, Dashboard Admin :
Fragment Permintaan Pinjaman**

No.	Aksi	Hasil	Keterangan
1	Tap Permintaan Pinjaman	Tampilkan Detail Permintaan Pinjaman	✓
2	Tap Identitas Peminjam	Tampilkan Detail Identitas Peminjam	✓
3	Dalam <i>Activity</i> Detail Peminjam : Tap Batalkan Pinjaman	Permintaan Pinjaman Dibatalkan Peminjam Diminta Melakukan Perbaikan Biodata	✓
4	Tap Setujui Pinjaman	<i>Update</i> Pinjaman Menjadi Siap Didanai Pendana	✓

**TABEL 4.14. Pengujian Blackbox, Dashboard Admin :
Fragment Pendanaan Pinjaman**

No.	Aksi	Hasil	Keterangan
1	Tap Pinjaman Aktif	Tampilkan Detail Pinjaman Aktif	✓
2	Tap Pinjaman dengan Pendanaan belum diverifikasi	Tampilkan Detail Permintaan Pendanaan	✓
3	Dalam <i>Activity</i> Detail Pinjaman belum diverifikasi : Tap Lihat Peminjam	Tampilkan Data Peminjam	✓
4	Dalam <i>Activity</i> Detail Pinjaman belum diverifikasi : Tap Lihat Pendana	Tampilkan Data Pendana	✓

**TABEL 4.14. Pengujian *Blackbox*, *Dashboard Admin* :
Fragment Pendanaan Pinjaman (Lanjutan)**

No.	Aksi	Hasil	Keterangan
5	Dalam <i>Activity</i> Detail Pendana : Tap Batalkan Pendanaan	Permintaan Pendanaan dibatalkan. Pendana diminta Melakukan Perbaikan Biodata sebelum melakukan pendanaan kembali.	✓
6	Dalam <i>Activity</i> Detail Pinjaman belum diverifikasi : Konfirmasi Pendanaan	<i>Update</i> Pinjaman menjadi Pinjaman Aktif	✓
7	Tap Pinjaman dengan Pinjaman Aktif	Tampilkan Detail Pinjaman Aktif	✓
8	Dalam <i>Activity</i> Detail Pinjaman Aktif : Tap Detail Peminjam	Tampilkan Data Peminjam	✓
9	Dalam <i>Activity</i> Detail Pinjaman Aktif : Tap Detail Pendana	Tampilkan Data Pendana	✓

**TABEL 4.15. Pengujian *Blackbox*, *Dashboard Admin* :
Fragment Cicilan Pinjaman**

No.	Aksi	Hasil	Keterangan
1	Tap Cicilan Pinjaman	Tampilkan Detail Pinjaman	✓
2	Dalam <i>Activity</i> Pembayaran Cicilan : Tap Konfirmasi Transfer	Transfer Cicilan dikonfirmasi	✓

**TABEL 4.16. Pengujian *Blackbox*, *Dashboard Admin* :
*Fragment Admin***

No.	Aksi	Hasil	Keterangan
1	Tap <i>Logout</i>	Keluar dari <i>Login</i> Admin	✓



BAB V

PENUTUP

5.1. KESIMPULAN

Berdasarkan hasil yang didapat dari pembuatan Rancang Bangun Aplikasi Peminjaman Uang “Pinjam Aja!” dengan model *P2P Lending*, maka dapat ditarik kesimpulan untuk menjawab rumusan masalah, diantaranya sebagai berikut :

1. Perancangan aplikasi menggunakan UML, dengan diagram yang dipakai dalam desain sistem adalah Diagram *Use Case*, Diagram *Activity* dan Diagram *Class*.
2. Perancangan awal tampilan (*layouting*), termasuk didalamnya perencanaan tata letak, desain ikon dan gambar yang digunakan menggunakan kombinasi dari aplikasi *Adobe Illustrator*, *Adobe Photoshop* dan *Adobe XD*.
3. Fungsi dari aplikasi dibangun menggunakan bahasa pemrograman Java, kemudian XML pada bagian tampilan (*layout*) dengan menggunakan IDE (*Intergrated Development Enviroment*) *Android Studio*. Sedangkan pada basis data menggunakan tipe *NoSQL* berbasis *Document* dari *Google Firebase*.
4. Untuk menguji fungsionalitas dari aplikasi sudah berjalan sesuai dengan hasil yang di harapkan, dilakukan pengujian menggunakan metode *Blackbox testing*.

5.2. SARAN

Untuk penelitian lebih lanjut dari aplikasi ini, bisa dilakukan penerapan aplikasi dalam transaksi yang nyata dan menambahkan fitur-fitur pendukung dalam kenyamanan bertransaksi melalui platform ini.

DAFTAR PUSTAKA

- AFPI. (2019). Pedoman Perilaku Pemberian Layanan Pinjam Meminjam Uang Berbasis Teknologi Informasi Secara Bertanggung Jawab. Asosiasi FinTech Pendanaan Bersama Indonesia.
<https://afpi.or.id/detailsnews/pedoman-perilaku> Diakses 11 Januari 2020 pada 10:54 WIB.
- Anonym. (2019). *P2P Lending*. http://id.wikipedia.org/wiki/P2P_Lending
Diakses tanggal 24 September 2019 pada 02:36 WIB.
- Bachmann, Alexander, et al. (2011). *Journal of Internet Banking and Commerce: Online Peer-to-Peer Lending – A Literature Review*. Array Development: Luneburg.
- Balyuk, Tetyana. (2019). *Financial Innovation and Borrowers: Evidence from Peer-to-Peer Lending*. Goizueta Business School, Emory University.
- B-j. Ma, Z-l. Zhou, F-y. Hu. (2017). *Pricing Mechanisms in the Online Peer-to-Peer Lending Market, Electronic Commerce Research and Applications*.
doi: <https://doi.org/10.1016/j.elerap.2017.10.006>
- Firebase. (2020). *Cloud Storage*. <https://firebase.google.com/docs/storage>
Diakses tanggal 5 Maret 2020 pada 20:05 WIB.
- Firebase. (2020). Model Data Cloud Firestore.
<https://firebase.google.com/docs/firestore/data-model> Diakses tanggal 28 Februari 2020 pada 00:03 WIB.
- Fitzgerald, Michael. (2012). *Introducing Regular Expression*. O'Reilly Media: Sebastopol.
- Jackson, Wallace. (2017). *Android Apps for Absolute Beginners: Covering Android 7*. Apress Media, LLC: California.
- Jonathan. (2018). Memahami Syarat Koperasi Pinjaman.
<https://koinworks.com/blog/memahami-syarat-koperasi-pinjaman/>
Diakses tanggal 28 September 2019 pada 19:24 WIB.
- Leavitt, Neal. (2010). *Will NoSQL Database Live Up to Their Promise?*. IEEE Computer Society.
- Mohan, C. (2013). *History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla*. Proc. 16th Int'l Conf. on Extending Database Technology: San Jose.

Monorey, Laurence. (2017). *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform*. Apress Media, LLC: California.

Munawar. (2005). *Pemodelan Visual dengan UML*. GRAHA ILMU: Yogyakarta.

OJK. (2016). Salinan Peraturan Otoritas Jasa Keuangan Nomor 77/POJK.01/2016 Tentang Layanan Pinjam Meminjam Berbasis Teknologi Informasi. <https://www.ojk.go.id/id/regulasi/otoritas-jasa-keuangan/peraturan-ojk/Documents/Pages/POJK-Nomor-77-POJK.01-2016/SAL%20-%20POJK%20Fintech.pdf> Diakses pada 7 Januari 2020 pada 13:48 WIB.

Pressman, Roger. (2012). *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7*. Yogyakarta: Andi.

Pressman, Roger S. (2005). *Software Engineering*. McGraw Hill: New York.

S. Cai, et al., (2016). *Judging online peer-to-peer lending behavior: A comparison of first-time and repeated borrowing requests*, Inf. Manage., <http://dx.doi.org/10.1016/j.im.2016.07.006>

Siregar, Helmi F., Nilfa Sari. (2018). *Rancang Bangun Aplikasi Simpan Pinjam Uang Mahasiswa Fakultas Teknik Universitas Asahan Berbasis Web*. Jurnal Teknologi Informasi.

