

Perancangan Aplikasi Pembukuan Menggunakan Metode *Agile Scrum*

<http://dx.doi.org/10.28932/jutisi.v8i3.5060>

Riwayat Artikel

Received: 06 Juli 2022 | Final Revision: 12 Oktober 2022 | Accepted: 18 Oktober 2022

Creative Commons License 4.0 (CC BY – NC)



Dhandy Joenathan Kurnia Putra[✉]#1, Penidas Fiodinggo Tanaem^{#2}

[#] Jurusan Sistem Informasi, Universitas Kristen Satya Wacana
Jl. Diponegoro 52-60, Salatiga, 50711, Indonesia

¹682018034@student.uksw.edu

²penidas.fiodinggo@uksw.edu

[✉]Corresponding author: 682018034@student.uksw.edu

Abstrak — Proses bisnis pada bisnis toko Sepatu Kaki Kaki pada bagian produk dan manajemen transaksi belum memaksimalkan sistem aplikasi yang ada. Hal ini membuat proses bisnis yang dilakukan oleh toko menjadi kurang efisien. Metodologi yang digunakan yaitu Scrum, salah satu model agile. Kelebihannya antara lain persyaratan yang fleksibel, waktu produksi aplikasi yang lebih cepat, dan anggota tim yang tidak banyak. Tujuan dari penelitian ini adalah untuk mengelola produk secara efektif dan membantu menganalisis penjualan berdasarkan tujuan yang diharapkan di masa depan. Bahasa pemrograman Kotlin - Android Native dengan BaaS (*Backend as a Service*) Firebase digunakan untuk mengembangkan aplikasi ini. Adapun penelitian ini akan menghasilkan sebuah aplikasi android yang mempercepat dan mempermudah pengelolaan produk, mengetahui persediaan produk terbaru.

Kata kunci — agile; android; firebase; scrum; software development.

Developing Bookkeeping Applications Using the Agile Scrum Method

Abstract — The product and transaction management section's business process in the Sepatu Kaki Kaki has not maximized the existing system application. This makes the business processes carried out by the store less effective. The method used in this study uses one of the agile software development models, namely Scrum. Which has advantages, including flexible requirements, speed up application production time, and require fewer team members. This research aims to produce application development that can manage products effectively and help staff analyze in accordance with the expected targets in the future. The application development uses the Kotlin programming language - Android Native, with BaaS (*Backend as a Service*) Firebase. The results of this research are an android application to speed up and simplify product management, the latest product stock can be known, and minimize manual recording errors.

Keywords — agile; android; firebase; scrum; software development.

I. PENDAHULUAN

Dalam dunia digital saat ini, peran teknologi sangat penting karena dapat menjadi katalisator perubahan mendasar dalam struktur, operasi, dan manajemen perusahaan. Hal ini dapat meningkatkan produktivitas, mengurangi biaya, meningkatkan pengambilan keputusan, meningkatkan hubungan pelanggan dan mengembangkan aplikasi strategis baru [1] [2].

Sepatu Kaki Kaki adalah nama dari usaha toko yang bergerak di bidang usaha penjualan sepatu dan sandal. Saat ini, usaha ini sudah memiliki 2 cabang toko yang dimana cabang pertama berada di Jl. Kemiri dan untuk cabang kedua berada di Jl. Tegalarjo. Usaha ini sudah mulai dirintis sejak tahun 2015 hingga saat ini. Setiap tahunnya terlihat bahwa ada perkembangan yang cukup signifikan mulai dari produk, jumlah pembelian, dll.

Khusus untuk pembukuan produk, sebagaimana yang dilakukan oleh para karyawan, saat ini secara umum proses pembukuan masih dilakukan secara manual. Mulai dari proses pencatatan keluar masuknya produk, ketersediaan stok produk, retur, hingga proses terjadinya transaksi. Hal ini dapat menyebabkan produktivitas menurun, tidak efektif dan kasus *human error* meningkat, sehingga dapat berpengaruh negatif terhadap pertumbuhan bisnis toko.

Berdasarkan pada permasalahan yang telah diuraikan, maka diperlukan suatu inovasi guna meminimalisir terjadinya masalah tersebut yaitu perancangan aplikasi pembukuan sebagai penunjang aktivitas bisnis pada toko Sepatu Kaki Kaki. Tujuan dari penelitian ini adalah untuk mengelola data produk baik itu produk yang keluar atau produk yang masuk pencatatan transaksi beserta retur secara otomatis, serta dapat membuat laporan transaksi secara sistematis. Tujuan lainnya adalah dengan adanya aplikasi ini dapat mempercepat kinerja karyawan menjadi lebih akurat dan meminimalisir adanya *human error*.

Untuk mencapai tujuan tersebut tentunya dalam merancang aplikasi diperlukan metode pengembangan sistem. Salah satunya adalah *Waterfall* yang merupakan metode pengembangan sistem yang sering digunakan, namun metode ini kurang efektif untuk diterapkan karena siklus pengembangan tidak memungkinkan adanya perubahan dan membutuhkan waktu yang lebih lama serta tim yang lebih besar. Karena tingkat kerumitan metode ini yang tinggi, mungkin akan sulit menggunakan metode pengembangan sistem *waterfall*. Karena dalam metode *waterfall*, tidak bisa kembali ke tahap sebelumnya / tidak fleksibel. Tidak seperti *waterfall*, *scrum* mampu beradaptasi dengan perubahan yang terjadi selama pengembangan aplikasi / fleksibel dan tidak membutuhkan anggota tim sebanyak metode *waterfall* [3]. *Scrum* merupakan metode pengembangan sistem yang banyak digunakan karena produktivitasnya yang tinggi [4]. Melihat dari penjelasan tersebut, maka pada penelitian ini akan menggunakan metode *Scrum*. Yang bertujuan untuk menghasilkan pengembangan aplikasi dengan cepat, dapat menyesuaikan perubahan, serta dapat sesuai dengan target yang diharapkan.

Terdapat penelitian-penelitian sebelumnya yang dijadikan sebagai acuan dalam penelitian ini. Penelitian sebelumnya yang pertama terkait dengan perancangan aplikasi dilakukan oleh Ganda Wijaya dan Melza Sari (2017) dengan judul Perancangan Sistem Informasi Aplikasi Kredit Berbasis Web Pada PT. BPR Mandiri Kredit Indonesia Cabang Bekasi. Berdasarkan permasalahan yang ada, tujuan yang ingin dicapai adalah mengatasi kendala yang ada, seperti lambatnya proses *history* kredit. Dari apa yang telah dicapai, telah dirancang sebuah aplikasi pengajuan kredit yang dapat mempermudah proses pengajuan kredit. Yang dimana pemohon tidak perlu hadir secara fisik untuk menyelesaikan proses aplikasi kredit, melainkan dapat dilakukan dan diakses dimana saja [5].

Penelitian terdahulu selanjutnya, dirancang oleh Gilang Pamungkas dan Herman Yuliansyah yang judul Merancang dan Membangun Aplikasi Cafe Android POS (*Point of Sale*) Untuk Kasir *Portable* dan *Printer Bluetooth* (2017). Adapun hasil yang dirancang adalah sebuah aplikasi kasir *portable* untuk proses transaksi penjualan, mengganti kasir *coffee shop* secara manual merangkum pendapatan ke dalam *file excel* dan cetak bukti pembayaran atau struk yang sudah terhubung ke *printer bluetooth*. Penelitian ini menjanjikan untuk mengurangi risiko kesalahan manusia [6].

Penelitian terdahulu yang terakhir adalah dari Nelly Monica, Sumitro Sarkum, dan Iwan Purnama dengan judul Aplikasi Data Mahasiswa Berbasis Android: Studi pada Sekolah Tinggi Ilmu Ekonomi Labuhanbatu. Adapun tujuan dari penelitian ini adalah membantu staff maupun dosen melakukan kegiatan operasional agar dalam hal pengolahan data mahasiswa dapat dilakukan secara cepat, baik dalam penginputan data mahasiswa hingga pencarian data mahasiswa. Terdapat hasil dari penelitian ini adalah terciptanya aplikasi data mahasiswa yang memanfaatkan database sebagai tempat penyimpanan data sekaligus dapat terkomputerisasi secara cepat, akurat, optimal dan aman [7].

Mengacu pada penelitian-penelitian terdahulu, pada penelitian ini akan berfokus pada penggunaan aplikasi berbasis android yang terintegrasi dengan printer *bluetooth* kasir sebagai pencetakan dan bukti transaksi. Adapun penyimpanan data yang digunakan adalah memanfaatkan *cloud storage* milik perusahaan *Google* yang bernama *Firebase*. Salah satu alasan memilih *Firebase* yaitu layanan yang ditawarkan sudah cukup lengkap, terdapat layanan *Authentication*, *Firestore Database*, hingga *Storage*.

II. METODE PENELITIAN

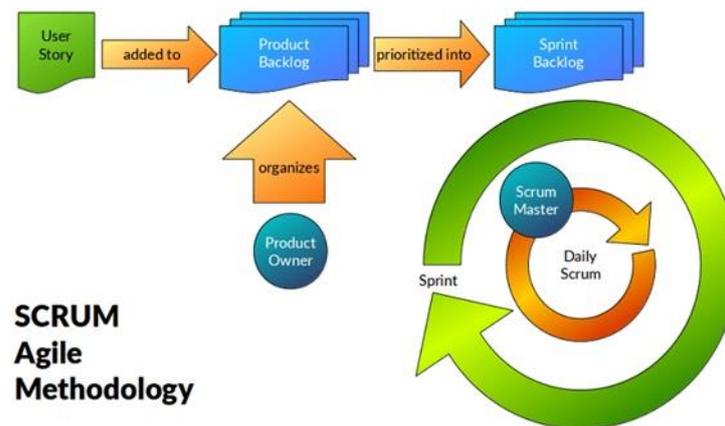
A. Agile

Pendekatan untuk pengembangan perangkat lunak yang cepat karena persyaratan berubah dalam waktu yang relatif singkat merupakan pengertian *Agile*. Pendekatan ini juga populer saat ini karena memberikan fleksibilitas terhadap pengembang ketika melakukan proses pengembangan [8]. Konsep utama pengembangan tangkas adalah pengembangan aplikasi, kolaborasi, dan komunikasi antar tim. Fokus pada pekerjaan aplikasi dengan meminimalkan dokumentasi. Bekerja sebagai tim yang terdiri dari 2 atau lebih *programmer* yang mengerjakan fitur dan komunikasi intensif antara *programmer* dan pelanggan. Praktik pengembangan *agile* cocok untuk proyek kecil dan dilakukan oleh tim kecil [9].

Ada beberapa macam metode *agile*, diantaranya adalah *Extreme Programming*, *Adaptive Software Development*, *Dynamic Systems Development Method* (DSDM) dan *Scrum* [10] [11]. Terdapat beberapa prinsip yang berlaku untuk mengimplementasikan pengembangan perangkat lunak [12], yakni: kepuasan pelanggan adalah prioritas utama, menerima perubahan *requirement*, bahkan di akhir pengembangan, memberikan hasil/perangkat lunak dalam beberapa minggu hingga bulan, selama proses pengembangan, lingkungan tim yang dapat dipercaya dan memotivasi satu anggota yang lain, mengedepankan komunikasi antar tim dan mengedepankan hasil fungsi dari *software* tersebut.

B. Scrum

Rekayasa perangkat lunak yang menerapkan prinsip-prinsip cepat dan mengandalkan sumber daya tim untuk mencapai hasil akhir merupakan pengertian dari *Scrum*. Menurut Schwaber & Sutherland, *scrum* adalah *framework* yang mampu menyelesaikan masalah kompleks yang terus berubah dan juga dianggap mampu secara kreatif dan produktif menghadirkan produk berkualitas tinggi sesuai keinginan pengguna [13]. Dalam teknik *scrum* ada tiga peranan penting, diantaranya *Product Owner* adalah yang bertanggung jawab atas produknya dan mengatur tim agar sejalan dengan visi, sehingga menghasilkan produk yang maksimal; *Scrum Master* bertindak sebagai pemilik produk dan fasilitator untuk tim pengembangan *Quality Assurance*. *Scrum Master* membantu tim mencapai tujuan dan mencari solusi jika ada hambatan pada *sprint*; dan yang terakhir *development Team/Scrum Team* mengembangkan tiket dalam *sprint*. Tim pengembangan atau tim pengembangan memiliki sekitar dua hingga sembilan anggota [14].



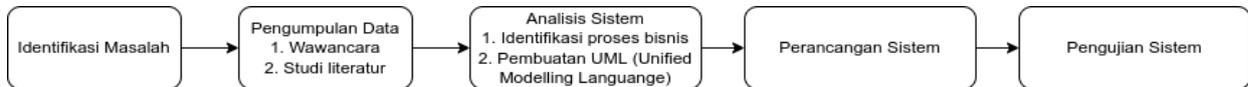
Gambar 1. Metode Pengembangan Sistem Agile Scrum [15]

Adapun penjelasan proses *scrum* secara umum dari Gambar 1 adalah sebagai berikut [16]:

- 1) *User Story* digunakan untuk menjelaskan yang ingin dicapai dan hasil yang diharapkan oleh pengguna. *User story* tersebut akan digunakan sebagai acuan untuk proses selanjutnya.
- 2) *Product backlog* merupakan daftar terurut dari semua yang dibutuhkan sistem atau produk. Isi *product backlog* dapat diubah untuk menghasilkan produk yang matang. Orang yang bertanggung jawab atas *product backlog* adalah pemilik produk.
- 3) *Sprint* adalah waktu pengembang mengerjakan *product backlog* dan berlangsung sekitar kurang lebih 1 bulan. Durasi *sprint* tidak berubah selama proses pengembangan produk. Tujuan dari *sprint* adalah untuk menyelesaikan tugas dari *sprint backlog*.

4) *Daily Scrum* adalah aktivitas harian di dalam *sprint* yang dilakukan oleh tim yang bertujuan untuk mempresentasikan hasil yang telah dicapai di hari sebelumnya.

Tahapan proses penelitian metode *Scrum* ini dapat dilihat pada Gambar 2 :



Gambar 2. Diagram Metode Penelitian

- Identifikasi masalah dilakukan untuk mengetahui permasalahan yang sedang terjadi pada aktivitas bisnis di toko Sepatu Kaki Kaki.
- Dalam penelitian ini, terdapat 2 langkah guna mengumpulkan data. Pertama, dengan melakukan wawancara terhadap karyawan toko, langkah ini bertujuan untuk mendapatkan informasi mengenai keperluan sistem yang dapat digunakan oleh toko serta menentukan tujuan yang ingin dicapai dari sistem yang akan dibangun. Kedua, dengan mempelajari literatur, guna untuk mengembangkan dan memperdalam topik yang akan dibahas dalam penelitian ini.
- Setelah melakukan pengumpulan data, maka bisa dilanjutkan dengan menganalisa dari hasil tersebut dengan merancang sistem menggunakan UML (*Unified Modeling Language*)
- Dari hasil analisa yang telah dirancang, maka dilanjutkan dengan perancangan sistem. Pada tahap perancangan sistem ini menggunakan metode *Agile Scrum*.
- Setelah perancangan sistem telah selesai dibuat, kemudian dilakukan pengujian sistem menggunakan metode *Black Box Testing* untuk memvalidasi fungsional yang telah dibuat sudah berjalan dengan baik dan sesuai dengan rancangan.

III. HASIL DAN PEMBAHASAN

A. User Story

Langkah pertama yang harus dikerjakan adalah *user story*. *User story* dibuat dengan bahasa pengguna secara umum. Hal ini berfungsi agar mempermudah memahami dari orang bisnis maupun orang teknis.

1) User Story Owner

- Sebagai seorang *owner*, saya ingin dapat melihat *history* transaksi dan retur dari setiap cabang.
- Sebagai seorang *owner*, saya ingin melihat grafik penjualan per bulannya di setiap cabang.
- Sebagai seorang *owner*, saya ingin dapat mencetak laporan transaksi dari masing-masing cabang.

2) User Story Karyawan

- Sebagai seorang karyawan, saya ingin mengelola data dari produk yang akan dijual.
- Sebagai seorang karyawan, saya ingin dapat menyimpan jumlah stok produk terkini.
- Sebagai seorang karyawan, kinerja saya ingin lebih efektif, bisa mencetak nota secara otomatis.
- Sebagai seorang karyawan, saya ingin dapat melakukan retur yang sudah terkomputerisasi.

B. Product Backlog

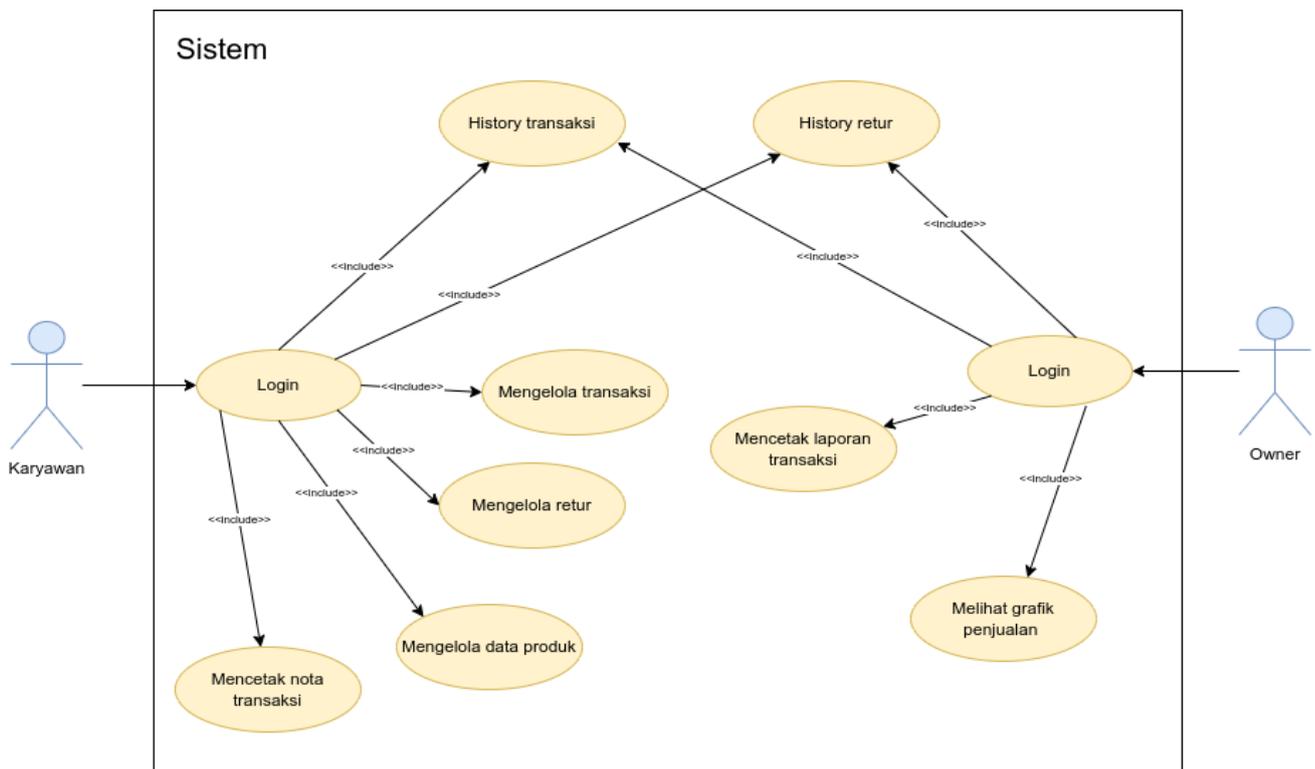
Merupakan tahap dimana hasil *user story* yang sudah diidentifikasi tersebut, dikelompokkan dan ditentukan prioritasnya. Tahap ini menghasilkan berupa hal seperti proses bisnis, aktor, dan struktur sistem. Berikut pada Tabel 1 merupakan *product backlog* yang dirancang berdasarkan analisa dari *User Story*.

TABEL 1
PRODUCT BACKLOG

No.	Fitur	Priority
1	Perancangan sistem dan <i>database</i>	<i>High Priority</i>
2	<i>Login</i> multi-user + session	<i>Medium Priority</i>
3	Dashboard dan profil (Karyawan)	<i>Low Priority</i>
4	Dashboard dan profil (Owner)	<i>Low Priority</i>
5	Menu pengelolaan produk	<i>Medium Priority</i>
6	Menu pengelolaan transaksi	<i>High Priority</i>

No.	Fitur	Priority
7	Menu pengelolaan retur	High Priority
8	Cetak nota (printer-bluetooth)	High Priority
9	Cetak laporan (soft-file PDF)	Medium Priority
10	Grafik penjualan	Medium Priority
11	Logout user	Low Priority

Selanjutnya di rancanglah UML (*Unified Modeling Language*). Berikut jenis UML yang dirancang antara lain *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*. *Use Case Diagram* digunakan untuk mengidentifikasi aktor-aktor yang akan berperan dalam aplikasi dan menggambarkan peran dari setiap aktor seperti pada Gambar 3.

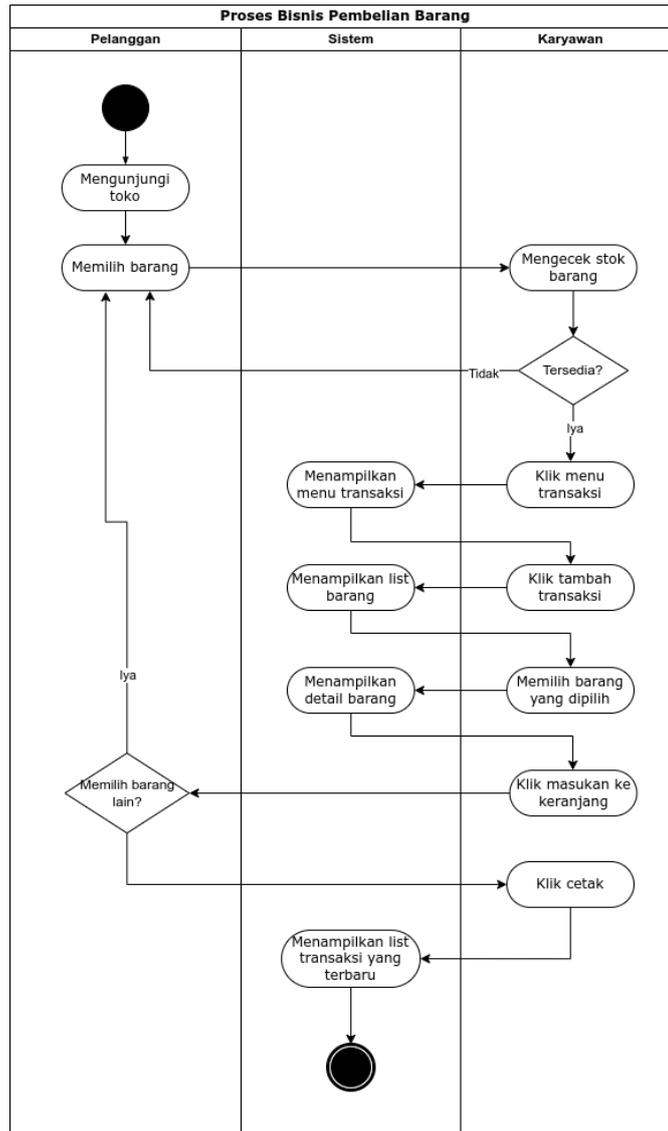


Gambar 3. Use Case Diagram

Berdasarkan pada Gambar 3. Aktor terbagi menjadi 2 aktor dengan aktivitas yang dapat dilakukan oleh masing-masing aktor. Berikut adalah penjelasan aktivitas yang dapat dilakukan oleh masing-masing aktor tersebut:

- Karyawan harus melakukan login terlebih dahulu, mencari produk yang dituju, mengelola data produk, mengelola transaksi, mengelola retur, dapat mencetak nota transaksi via *printer bluetooth*, serta dapat melihat *history* transaksi
- Owner dapat menampilkan *history* penjualan dari setiap cabang baik transaksi maupun retur yang terjadi, dapat mencetak laporan transaksi dan retur, melihat grafik penjualan tahunan di setiap bulannya. Semua aktivitas dari aktor tersebut memerlukan proses autentikasi *login*.

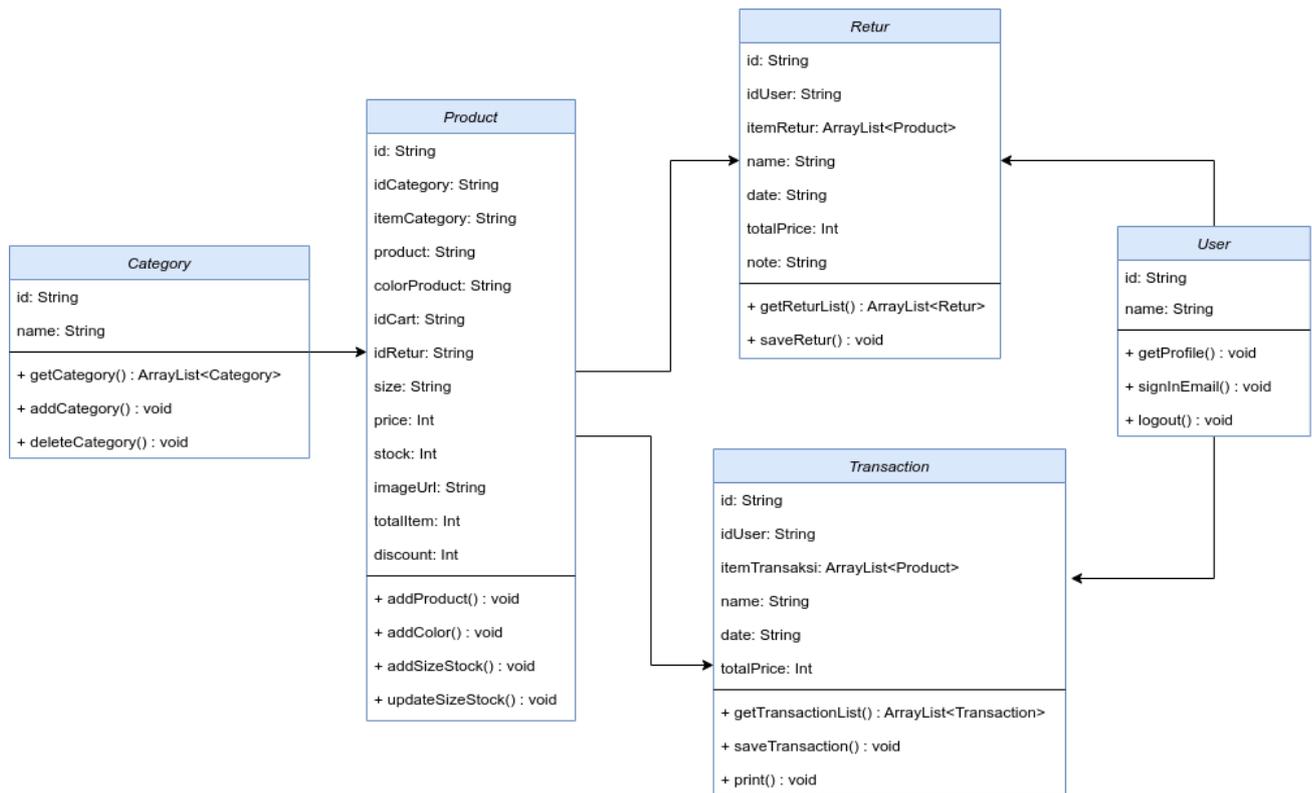
Setelah mengidentifikasi aktor yang berperan dalam sistem, tahapan selanjutnya adalah mengidentifikasi proses bisnis yang akan dijalankan menggunakan activity diagram. Pada Gambar 4 dijelaskan urutan aktivitas bisnis yang terjadi berdasarkan diagram use case yang telah dibuat.



Gambar 4. Activity Diagram

Bisa dilihat pada Gambar 4, terbagi menjadi 3 *swimlane* yaitu Pelanggan, Sistem, dan Karyawan. Dimulai dari pelanggan datang ke toko memilih barang, lalu karyawan akan melakukan pengecekan stok. Jika tersedia, karyawan bisa membuka menu transaksi di aplikasi dan mencari barang tersebut untuk dimasukkan ke fitur keranjang. Jika Pelanggan ingin membeli lebih, maka proses kembali ke awal. Setelah Pelanggan selesai memilih barang, karyawan bisa menekan tombol Cetak di aplikasi. Dan secara otomatis sistem akan menyimpan data transaksi tersebut dan melakukan cetak nota via *printer bluetooth*. Setelah selesai, sistem akan menampilkan *list* transaksi terbaru dan aktivitas bisnis selesai.

Class Diagram memberikan gambaran struktur sistem yang berisikan sekumpulan kelas dan menjelaskan gambaran terkait. Perancangan class diagram merupakan salah satu hal yang penting karena menjadi acuan dalam pembuatan sistem. Setelah mengidentifikasi model data yang sedang dikembangkan use case dan proses bisnis, class diagram dibuat untuk memperjelas kebutuhan sistem seperti pada Gambar 5.



Gambar 5. Class Diagram

Terdapat 5 class yang dibangun pada aplikasi sesuai Gambar 5. Adapun *Category*, *Product*, *Retur*, *Transaction*, dan *User* dengan peran yang berbeda-beda.

- *Class Category* : suatu model data yang digunakan untuk menyimpan kategori produk. Terdapat atribut id dan nama kategori tersebut.
- *Class Product* : suatu model data yang digunakan untuk menyimpan data produk. Dalam kelas ini, sekaligus memiliki atribut yang akan digunakan untuk keperluan transaksi dan retur, seperti atribut *totalItem* yang digunakan untuk menyimpan jumlah barang yang dibeli / retur.
- *Class Retur* : suatu model data yang digunakan untuk menyimpan informasi data retur. Seperti tanggal, total harga, note, item retur.
- *Class Transaction* : suatu model data yang digunakan untuk menyimpan informasi data transaksi. Seperti tanggal, total harga, item transaksi.
- *Class User* : suatu model data yang digunakan untuk menyimpan informasi *user*.

C. Sprint

Terdapat 4 tahap yaitu *sprint planning*, *sprint backlog*, *sprint execution*, *sprint retrospective*.

Sprint Planning: terdapat 4 fitur yang disepakati yaitu perancangan fitur *login*, *dashboard* dan profil karyawan beserta *owner*, fitur pengelolaan produk untuk karyawan, dan *logout user*. Setelah menentukan fitur *backlog*, selanjutnya masuk dalam tahap *Sprint Backlog*, yang dimana membagi *backlog* tersebut ke bagian-bagian yang lebih kecil.

Sprint Backlog: berisikan *backlog* yang diuraikan lebih rinci. Agar mempermudah tim pengembang dalam menyelesaikan sprint seperti pada Tabel 2.

TABEL 2
PRODUCT BACKLOG

No.	Fitur	Priority
1	<i>Login multi-user</i>	Tampilan sebelum masuk ke <i>dashboard</i> pengguna Terdapat <i>session</i> dimana sistem akan mengecek apabila sudah <i>login</i>

No.	Fitur	Priority
		sebelumnya, maka secara otomatis akan dialihkan ke dashboard pengguna
2	Logout user	Ketika melakukan <i>logout</i> , akan dialihkan ke halaman <i>login</i>
3	Dashboard pengguna sistem	Dashboard owner Dashboard karyawan
4	Mengelola data produk	Menambah data produk Menghapus data produk Mencari data produk Mengubah data produk

Sprint Execution: ketika *sprint backlog* sudah disetujui, maka bisa memulai *sprint execution*. Daftar fitur yang telah dibuat pada *sprint backlog* dikumpulkan untuk dikerjakan pada *sprint execution* seperti yang terlihat dalam Tabel 3.

TABEL 3
STATUS TASK BACKLOG PADA DAILY SCRUM

Fitur Backlog	Fitur	Waktu (Hari)	Status
Login multi-user	Tampilan utama sebelum masuk ke dashboard pengguna	2	Selesai
	Pengecekan <i>session</i> oleh sistem	1	
Logout user	Ketika melakukan <i>logout</i> , akan dialihkan ke halaman <i>login</i>	1	Selesai
Dashboard pengguna sistem	Dashboard owner	3	Selesai
	Dashboard karyawan	3	Selesai
Mengelola data produk	Menambah kategori / produk	2	Selesai
	Menghapus kategori / produk	1	Selesai
	Mencari kategori / produk	1	Selesai
	Mengubah kategori / produk	2	Selesai
Mengelola transaksi	List history transaksi	1	Selesai
	Detail transaksi	2	Selesai
	Fitur keranjang	7	Selesai
	Menyimpan data transaksi	3	
Mengelola retur	List history retur	1	Selesai
	Detail retur	2	Selesai
	Menyimpan data retur	3	Selesai
Cetak nota (printer bluetooth)	Pairing ke printer bluetooth	2	Selesai
	Generate data ke format nota	3	Selesai
Cetak laporan (PDF)	Cetak nota otomatis	1	
	Generate data ke format file PDF	2	Selesai
Grafik penjualan	Download laporan PDF	1	Selesai
	Generate data ke dalam bentuk grafik tabel	4	Selesai

Sprint Retrospective: dimana tahap ini dilakukan pada hari terakhir *sprint*. Di dalam pertemuan ini membahas seputar apa yang telah dicapai selama waktu *sprint* berlangsung, halangan yang terjadi saat pengerjaan *sprint* dan perencanaan mengenai peningkatan cara kerja tim yang akan dilakukan di *sprint* berikutnya agar *sprint* berikutnya lebih efektif.

D. Delivery Product



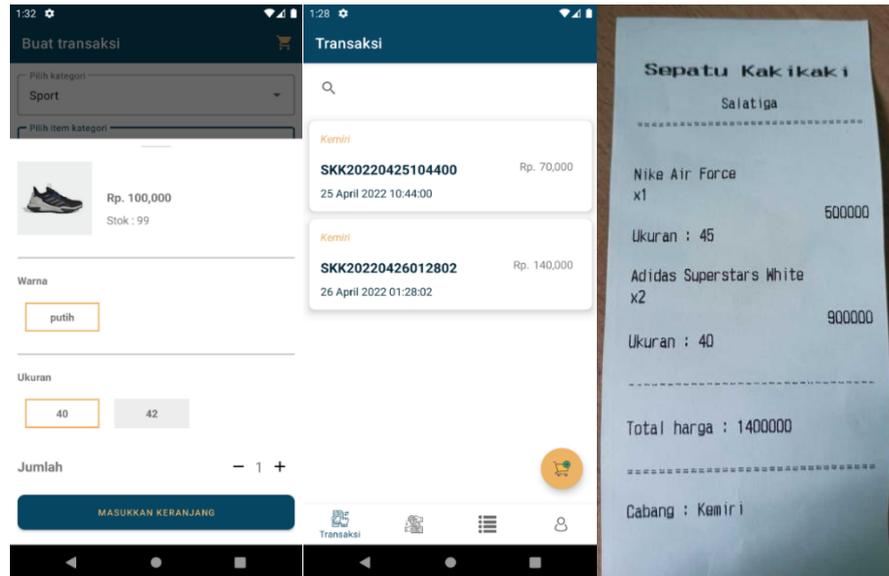
Gambar 6. Halaman *login* aplikasi

Gambar 6 merupakan halaman *login* dimana *user* harus *login* sebelum masuk ke aplikasi sesuai dengan *role* yang diinginkan, sebagai karyawan atau *owner*.

```
1 private fun signInEmail(email: String, password: String) {
2     binding.llLogin.visibility = View.VISIBLE
3     binding.pbLogin.visibility = View.VISIBLE
4     firebaseAuth.signInWithEmailAndPassword(email, password)
5         .addOnCompleteListener(this) { task ->
6             if (task.isSuccessful) {
7                 // Sign in success, update UI with the signed-in user's information
8                 binding.pbLogin.visibility = View.GONE
9                 binding.llLogin.visibility = View.GONE
10
11                 if (email == "owner@sepatukakikaki.com") {
12                     val intent = Intent(this, OwnerActivity::class.java)
13                     startActivity(intent)
14                     finish()
15                 } else {
16                     val intent = Intent(this, MainActivity::class.java)
17                     startActivity(intent)
18                     finish()
19                 }
20             }
21         }
22     .addOnFailureListener
23         Toast.makeText(this, "Gagal", Toast.LENGTH_SHORT).show()
24         binding.pbLogin.visibility = View.GONE
25         binding.llLogin.visibility = View.GONE
26     }
27 }
```

Kode Sumber 1. *Function login* aplikasi beserta pengecekan *role* yang akan digunakan, apakah *role* karyawan / *owner*.

Pada kode sumber 1 merupakan tampilan *source code* fungsi *login* aplikasi. Dengan menambahkan fungsi pada baris 4, sistem akan secara otomatis mengirimkan *request* ke *backend Firebase* untuk mengambil data. Dan pada baris ke 11-18 sistem akan memvalidasi *role user* sesuai dengan email yang dimasukkan, apakah sebagai karyawan atau *owner*. Setelah *login* berhasil, sistem akan memberikan akses *role* pada *user*.



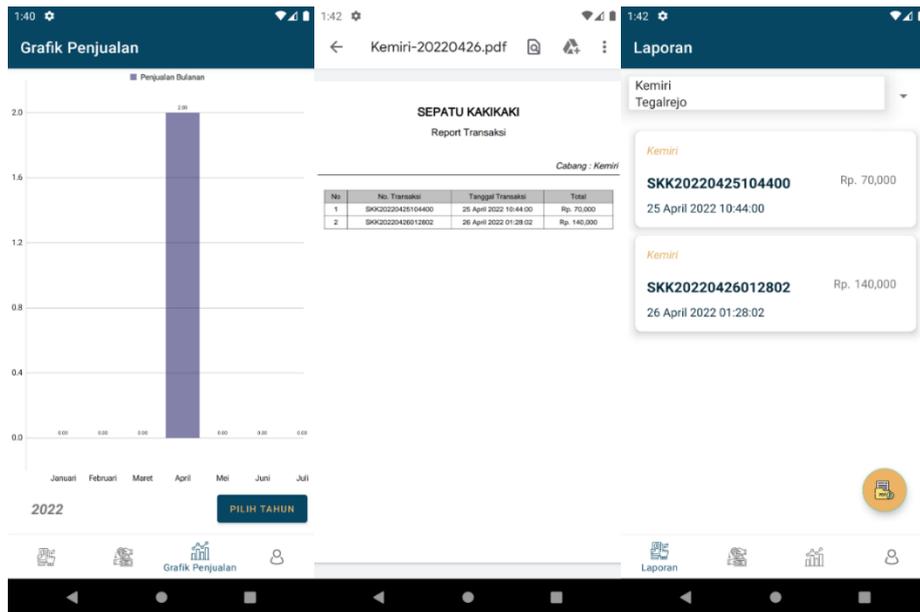
Gambar 7. Halaman aplikasi *role* Karyawan

Gambar 7 memperlihatkan halaman yang nantinya akan diakses oleh *role* karyawan. Dimana di halaman *dashboard* tersebut terdapat 3 fitur penting yaitu pengelolaan transaksi, pengelolaan produk, dan cetak nota otomatis via *printer bluetooth*. Adapun menu *profile* untuk melakukan *logout*. Selain itu, produk yang tersimpan dalam *database* dapat ditampilkan dengan baik.

```
1 private fun saveTransaction(dataitem: ArrayList<Item>) {
2     val patternNameTransaction = "yyyyMMddHHmm"
3     val simpleDateFormat1 = SimpleDateFormat(patternNameTransaction)
4     val nameTransaction: String = simpleDateFormat1.format(Date())
5
6     val patternDateTransaction = "dd MMMM yyyy hh:mm:ss"
7     val simpleDateFormat2 = SimpleDateFormat(patternDateTransaction, Locale("ID"))
8     val dateTransaction: String = simpleDateFormat2.format(Date())
9
10    // save transaction
11    val docTransaction = firebaseDB.collection(COLLECTION_USERS)
12        .document(currentUser?.uid ?: "")
13        .collection(COLLECTION_TRANSACTION).document()
14    val item = Transaction(docTransaction.id, "transaksi-$nameTransaction",
15        convertUidToName(currentUser!!.uid), dateTransaction, totalPrice)
16    docTransaction.set(item)
17
18    for (i in dataitem.indices) {
19        database.saveTransactionItem(currentUser?.uid ?: "", dataitem[i],
20            docTransaction.id)
21        database.updateStockItem(currentUser?.uid ?: "", dataitem[i])
22    }
23}
```

Kode Sumber 2. *Function* untuk menyimpan transaksi.

Kode sumber 2 merupakan tampilan *source code* fungsi untuk penyimpanan transaksi. Sistem akan secara otomatis melakukan *request* ke *backend Firebase* dan menyimpan data transaksi beserta *field* seperti tanggal transaksi, nama transaksi dan item transaksi pada baris ke 11-21



Gambar 8. Halaman aplikasi *role Owner*.

Gambar 8 memperlihatkan halaman aplikasi yang akan diakses oleh *role owner*. Dimana terdapat 2 fitur penting yaitu grafik penjualan selama setahun setiap bulannya dan cetak *soft-file* pdf transaksi dari masing-masing cabang. *User* bisa melihat transaksi apa saja yang terjadi dari masing-masing cabang pada menu laporan. Adapun menu *profile* untuk melakukan *logout*.

```
1 private fun printPDF(branch: String, data: ArrayList<Transaction>) {
2     // formatting name PDF
3     val patternNameTransaction = "yyyyMMdd"
4     val simpleDateFormat1 = SimpleDateFormat(patternNameTransaction)
5     val nameDocument: String = simpleDateFormat1.format(Date())
6
7     val file = File(thisContext.getExternalFilesDir("/"), "$nameDocument.pdf")
8
9     val writer = PdfWriter(file)
10    val pdfDocument = com.itextpdf.kernel.pdf.PdfDocument(writer)
11    val document = Document(pdfDocument)
12    var counter = 1
13
14    // Table
15    val table = Table(floatArrayOf(40F, 180F, 160F, 120F))
16        .setHorizontalAlignment(HorizontalAlignment.CENTER)
17
18    // New line
19    val newline = Paragraph(Text("\n"))
20    val paragraph1 = Paragraph("SEPATU KAKIKAKI")
21        .setTextAlignment(TextAlignment.CENTER)
22    ...
23
24    // add layout in PDF
25    document.add(paragraph1)
26    document.add(paragraph2)
27    document.add(newline)
28    document.add(paragraph3)
29    document.add(line)
30    document.add(newline)
31    document.add(table)
```

```
32 document.close()
33 Toast.makeText(thisContext, "PDF berhasil dibuat!",
34 Toast.LENGTH_SHORT).show()
```

Kode Sumber 3. Generate file PDF

Bisa dilihat pada kode sumber 3 merupakan tampilan *source code* fungsi untuk men-generate file pdf guna menghasilkan *report history* transaksi. Sistem akan mengambil data dari *backend Firebase* lalu diolah sedemikian rupa sesuai dengan format yang diinginkan. Berawal dari pendefinisian *file* pada baris ke 3-12, lalu pendefinisian format *wording*, tabel, data yang ditampilkan pada baris ke 14-34.

E. Testing

Di dalam tahap *sprint execution*, terdapat tahap *testing* dimana melakukan tes terhadap fitur yang telah dibuat pada akhir *sprint*. Pada pengujian ini menggunakan *black box testing*, dimana dilakukan uji untuk mengetahui hasil *input output* dari aplikasi. Hasil *testing* dapat dilihat pada Tabel 4.

TABEL 4
STATUS TASK PADA DAILY SCRUM

No.	Skenario testing	Sistem respon yang diharapkan	Hasil
1.	Email beserta password yang dimasukkan benar	Tampilan berpindah ke <i>dashboard</i>	Sukses
2	Email beserta <i>password</i> yang dimasukkan salah	Tampilan tidak berpindah dan muncul pesan <i>error</i> berwarna merah	Sukses
3.	Penampilan <i>dashboard</i>	<i>Dashboard</i> yang muncul sesuai dengan akun / <i>role</i> masing-masing	Sukses
4.	Login <i>session</i> yang diterapkan pada sistem	Ketika pengguna sudah melakukan <i>login</i> sebelumnya, secara otomatis akan dialihkan ke halaman <i>dashboard</i> (tidak perlu <i>login</i> kembali)	Sukses
5.	<i>Logout user</i>	Halaman akan berpindah ke halaman <i>login</i>	Sukses
6.	<i>Profile user</i>	Berhasil menampilkan <i>profile user</i>	Sukses
6.	User dapat menambahkan, menghapus, mengubah dan mencari produk	Berhasil memasukan, mengurangi, mengubah, mencari barang,	Sukses
7.	Menyimpan data transaksi	Berhasil menyimpan data sesuai dengan data yang dipilih	Sukses
8.	Detail transaksi	Berhasil menampilkan detail transaksi	Sukses
8.	Menyimpan data retur	Berhasil menyimpan data sesuai dengan data yang dipilih	Sukses
8.	Detail retur	Berhasil menampilkan detail retur	Sukses
9.	<i>Pairing</i> ke printer <i>bluetooth</i>	Berhasil terhubung dengan printer <i>bluetooth</i>	Sukses
10.	Cetak nota transaksi menggunakan printer <i>bluetooth</i>	Berhasil mencetak nota transaksi menggunakan printer <i>bluetooth</i> dan format nota sesuai	Sukses
11.	Menampilkan <i>history</i> retur dari setiap cabang	Berhasil menampilkan semua <i>history</i> retur yang terjadi di setiap cabang	Sukses
12.	Menampilkan <i>history</i> transaksi dari setiap cabang	Berhasil menampilkan semua <i>history</i> transaksi yang terjadi di setiap cabang	Sukses
13.	Cetak laporan transaksi PDF	Berhasil men- <i>download</i> file PDF laporan transaksi	Sukses
14.	Menampilkan grafik penjualan	Berhasil menampilkan grafik penjualan selama setahun	Sukses

IV. SIMPULAN

Berdasarkan dari hasil penelitian yang sudah dilakukan, dihasilkan sebuah aplikasi yaitu aplikasi pembukuan berbasis android yang dapat mengelola pemasukan serta pengeluaran barang, dapat melakukan transaksi beserta cetak nota otomatis, sekaligus dapat melakukan pembuatan laporan penjualan secara otomatis. Sebelumnya pihak toko belum memiliki aplikasi

pendukung untuk mengontrol / memonitor stok barang beserta transaksi. Dengan adanya aplikasi ini dapat memudahkan pihak toko untuk mengontrol barang yang akan dijual. Lalu berdasarkan hasil dari pengujian sistem dengan menggunakan metode *black-box testing* dapat disimpulkan bahwa tujuan dalam penelitian ini berhasil dicapai guna menunjang aktivitas bisnis. Berdasarkan hasil wawancara dari salah satu karyawan toko, melalui aplikasi ini, karyawan dapat mengelola data produk dengan sistematis dan akurat, pencatatan transaksi beserta retur secara otomatis, dapat membantu membuat laporan transaksi secara sistematis dan aplikasi berjalan dengan baik.

Terdapat pengujian pula integrasi antara aplikasi dengan *printer bluetooth*. Dengan melihat hasil pengujian, dapat disimpulkan bahwa integrasi dengan *printer bluetooth* berjalan dengan baik. Aplikasi dapat terhubung / terputus dengan lancar. Hasil nota yang dicetak sudah sesuai dengan keinginan user, baik format maupun data yang ditampilkan. Adapun tantangan atau hambatan dari penggunaan fitur ini, yang pertama adalah ketika ingin menerapkan *multiple pairing* di 2 device yang berbeda. Dan yang kedua adalah ketika ingin melakukan cetak nota berjumlah dua / lebih secara bersamaan, ini disebabkan karena belum adanya sistem antrian / *queuing* di printer tersebut. Sehingga user diharuskan melakukan *request* cetak nota satu demi satu.

UCAPAN TERIMA KASIH

Ucapan terima kasih kepada toko Sepatu Kaki Kaki dan beberapa karyawan toko yang telah membantu dan bersedia untuk dijadikan tempat penelitian ini. Terimakasih kepada pembimbing, teman-teman yang sudah banyak membantu dalam pembuatan penelitian ini.

DAFTAR PUSTAKA

- [1] J. P. J. Prayoga, "Penerapan Teknologi Informasi Dalam Peningkatan Efektivitas, Efisiensi dan Produktivitas Perusahaan," *Warta Dharmawangsa*, no. 53, 2017.
- [2] B. Iskandar dan Hamdani, A. U, "Desain dan Pengujian Sistem Informasi Jasa Pengiriman Barang Studi Kasus: PT. XYZ," *Semnasteknomedia Online*, vol. 5, no. 1, pp. 1-2, 2017.
- [3] I. Kurniawan dan R. R. Sani, "Pemodelan SCRUM dalam Pengembangan Sistem Informasi Kesehatan pada Klinik Ar-Rokhim Sragen Kabupaten Sragen," *JOINS (Journal Inf. Syst.)*, vol. 4, no. 1, pp. 76-86, 2019.
- [4] S. Hardani, "Pengembangan Sistem Informasi KPR Syariah Dengan Metode Scrum," *J. Ilmu Pengetah. dan Teknol. Komput.*, vol. 4, no. 2, pp. 223-230, 2019.
- [5] G. Wijaya dan Sari, Melza, "Perancangan Sistem Informasi Pengajuan Kredit Berbasis Web Pada PT. BPR Kredit Mandiri Indonesia Cabang Bekasi," *Indonesian Journal on Software Engineering (IJSE)*, vol. 3, no. 2, 2017.
- [6] G. Pamungkas dan Yuliansyah, H, "Rancang Bangun Aplikasi Android Pos (Point of Sale) Kafe Untuk Kasir Portable dan Bluetooth Printer," *JST (Jurnal Sains dan Teknologi)*, vol. 6, no. 1, pp. 199-208, 2017.
- [7] N. Monica, Sarkum, S dan Purnama, I, "Aplikasi data mahasiswa berbasis android: studi pada Sekolah Tinggi Ilmu Ekonomi Labuhanbatu," *IT Journal Research and Development*, vol. 3, no. 1, pp. 43-53, 2018.
- [8] D. A. P. Putri, "Rancang Bangun Media Pembelajaran Bahasa Arab Untuk Anak Usia Dini Berbasis Android," *Technol. J. Ilm.*, vol. 10, no. 3, p. 156, 2019.
- [9] I. Larasati, Yusril, A dan Al Zukri, P, "Systematic Literature Review Analisis Metode Agile Dalam Pengembangan Aplikasi Mobile," *Sistemasi: Jurnal Sistem Informasi*, vol. 10, no. 2, pp. 369-380, 2021.
- [10] A. Anoesyirwan, H. Madiistriyatno dan S. Mutmainnah, "Peningkatan Kualitas Manajemen Publikasi Ilmiah Menggunakan Metode Agile," *ABDI*, vol. 1, no. 2, pp. 31-39, 2020.
- [11] I. Mahendra dan Yanto, D. T. E, "Sistem Informasi Pengajuan Kredit Berbasis Web Menggunakan Agile Development Methods Pada Bank Bri Unit Kolonel Sugiono," *Jurnal Teknologi Dan Open Source*, vol. 1, no. 2, pp. 13-24, 2018.
- [12] S. Pertamax, "Bedah 12 Prinsip Agile," 19 Maret 2019. [Online]. Available: <https://medium.com/au-ppl/bedah-12-prinsip-agile-98c756ecf0e4>. [Diakses 7 Februari 2022].
- [13] K. Schwaber dan J. Sutherland, "Panduan Scrum," November 2020. [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Indonesian.pdf>. [Diakses 15 Februari 2022].
- [14] A. D. L, "Metodologi Scrum," Desember 2019. [Online]. Available: <https://socs.binus.ac.id/2019/12/23/metodologi-scrum/>. [Diakses 5 Maret 2022].
- [15] L. Ihsan, K. Sulton dan R. Muhammad Rafli, "Software Engineering : Agile Software Development Process Model – Scrum Model," Februari 2022. [Online]. Available: <https://socs.binus.ac.id/2020/03/31/software-engineering-agile-software-development-process-model-scrum-model/>. [Diakses 26 Maret 2022].
- [16] A. Andipradana dan Hartomo, K. D, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *Jurnal Algoritma*, vol. 18, no. 1, pp. 161-172, 2021.