

## Model Aplikasi *Gym Member* dengan Pendekatan *Agile Scrum*

Julpandi<sup>1\*</sup>, Yuli Asriningtias<sup>2</sup>

Informatika, Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia  
 \*e-mail *Corresponding Author*: julpandi.5210411034@student.uty.ac.id

### Abstract

*The management of membership data and the registration process for members at AREA Fitness is still done manually, particularly with registrations being completed by filling out forms on-site, requiring customers to visit the location in person. This method is considered highly inefficient and prone to errors or data loss. Therefore, an Android-based Gym Member application has been developed to streamline the registration process, profile updates, and real-time membership data management. Firebase Realtime Database is implemented to ensure fast and efficient data synchronization. The Agile Scrum methodology was adopted to allow flexibility during the app development process. Firebase was chosen due to its real-time data management capabilities, supporting efficient membership data handling. The implementation testing results showed that the developed application helps make the membership data management and registration process more efficient. This implementation speeds up member registration and simplifies membership data management.*

**Key words:** *Firestore Realtime Database; Android; Agile Scrum; Member Registration*

### Abstrak

Pengelolaan data keanggotaan dan proses pendaftaran member gym di *AREA Fitness* masih dilakukan secara manual, terutama pendaftaran yang dilakukan dengan cara mengisi formulir di lokasi, sehingga *customer* harus datang langsung ke tempat tersebut. Hal ini dinilai sangat tidak efisien dan juga rentan terhadap kesalahan ataupun kehilangan data. Oleh karena itu, dikembangkanlah sebuah aplikasi *Gym Member berbasis Android* untuk mempermudah proses pendaftaran, pembaruan profil, serta pengelolaan data keanggotaan secara *real-time*. *Firestore Realtime Database* diimplementasikan untuk memastikan *sinkronisasi data* yang cepat dan efisien. Metode *Agile Scrum* digunakan untuk memungkinkan fleksibilitas dalam proses pengembangan aplikasi. *Firestore* dipilih karena kemampuannya mengelola data secara *real-time* dan mendukung efisiensi pengelolaan keanggotaan. Hasil uji implementasi menunjukkan bahwa aplikasi yang dibangun dapat membantu dalam proses pengelolaan data keanggotaan dan pendaftaran menjadi lebih efisien. Implementasi ini membuat proses pendaftaran anggota lebih cepat dan mempermudah pengelolaan data member.

**Kata kunci:** *Firestore Realtime Database; Android; Agile Scrum; Pendaftaran Member*

### 1. Pendahuluan

Pada era digital yang semakin maju, banyak bisnis, termasuk pusat kebugaran (gym), mulai memanfaatkan teknologi informasi untuk meningkatkan efisiensi operasional mereka. *AREA Fitness* saat ini masih mengelola data keanggotaan secara manual, yang melibatkan pelanggan untuk mengisi formulir pendaftaran fisik, sehingga menguras waktu dan tenaga baik bagi pelanggan maupun pengelola. Penggunaan sistem manual ini sering kali mengakibatkan kendala operasional, seperti potensi kehilangan data dan kesalahan pencatatan yang dapat menurunkan efektivitas pelayanan. Dalam studi yang dilakukan pada *Alto Myshouse*, sistem manual seperti ini terbukti memiliki kelemahan signifikan yang menghambat kecepatan akses data dan keakuratan informasi [1].

Pengelolaan keanggotaan di *AREA Fitness* saat ini masih dilakukan secara manual, pelanggan harus mengisi formulir fisik dan staf harus mengelola data pendaftaran secara terpisah, sehingga memakan waktu dan meningkatkan risiko kesalahan seperti kehilangan atau data yang kurang akurat yang dapat menghambat efektivitas layanan. Kondisi ini tidak sejalan dengan kebutuhan akan pengelolaan data yang cepat, akurat, dan *real-time* untuk mendukung

layanan yang efisien. Untuk mengatasi masalah tersebut, penggunaan aplikasi berbasis teknologi dengan *Firestore Realtime Database* sebagai backend menjadi solusi yang relevan. *Firestore* telah terbukti efektif dalam berbagai aplikasi yang membutuhkan sinkronisasi data *real-time*, seperti aplikasi *FeedbackMe* yang menghubungkan guru dan orang tua dalam pendidikan jarak jauh [2], serta aplikasi pemesanan jasa barbershop yang memudahkan pengelolaan layanan dan mengoptimalkan kinerja operasional [3]. Dengan *Firestore*, data keanggotaan dapat tersimpan dan tersinkronisasi secara otomatis, sehingga memungkinkan proses pendaftaran dan pengelolaan data keanggotaan di *AREA Fitness* menjadi lebih efisien, akurat, dan siap mendukung kebutuhan layanan cepat dan tepat.

Proses pengembangan perangkat lunak dalam aplikasi *Gym Member* membutuhkan fleksibilitas dan adaptasi tinggi untuk memenuhi kebutuhan pengguna yang selalu berubah. Metode *Agile Scrum* dipilih sebagai pendekatan utama karena memungkinkan pengembangan dilakukan dalam siklus yang pendek, sehingga dapat dengan cepat menyesuaikan fitur aplikasi sesuai kebutuhan pengguna [4]. *Agile Scrum* telah terbukti efektif dalam meningkatkan efisiensi dan adaptabilitas dalam berbagai sistem informasi, seperti *pengembangan sistem Point of Sales di Toko Eka Putra*, yang mengoptimalkan akurasi transaksi, serta *sistem pelaporan keuangan BUMDes* yang berhasil menerapkan penyesuaian berkelanjutan dalam setiap sprint [5]. Dengan membagi pengembangan aplikasi *Gym Member* menjadi *sprint-sprint* pendek, metode ini memungkinkan evaluasi dan pembaruan fitur secara terus-menerus, sehingga aplikasi dapat selalu relevan dengan kebutuhan pengguna. *Agile Scrum* diharapkan membantu proses pengembangan aplikasi *Gym Member* berjalan efisien dan juga mendukung pengelolaan keanggotaan secara *real-time*.

Tujuan penelitian ini adalah mengembangkan *aplikasi Gym Member berbasis Android* yang menggunakan *Firestore Realtime Database* dan metode *Agile Scrum* untuk meningkatkan efisiensi dalam proses pendaftaran serta pengelolaan data keanggotaan gym secara *real-time*. Aplikasi ini dirancang sebagai solusi terintegrasi yang tidak hanya mengotomatiskan pendaftaran anggota, tetapi juga memastikan sinkronisasi data secara cepat dan akurat antar perangkat. Berbeda dengan penelitian sebelumnya, aplikasi ini menawarkan kemampuan pengelolaan data keanggotaan yang lebih responsif, sehingga dapat membantu dalam memberikan layanan yang lebih efektif dan meminimalkan risiko kesalahan data.

## 2. Tinjauan Pustaka

Penelitian yang dilakukan oleh Marice Hotnauli Simbolon, Sulaiman, dan Pinondang Hutapea (2024) yang berjudul *Penerapan Metode Prototype pada Perancangan Aplikasi Pengelolaan Member Pusat Kebugaran*, mengembangkan aplikasi untuk membantu proses pengelolaan keanggotaan pusat kebugaran. Aplikasi ini mempermudah dalam pendaftaran dan perpanjangan member, penjadwalan latihan, serta pemilihan personal trainer. Dengan penerapan metode *Prototype*, aplikasi ini memungkinkan pemilik pusat kebugaran untuk lebih efisien dalam menyusun laporan dan mengelola kehadiran member. Melalui pengujian dan evaluasi prototipe, aplikasi dapat dioptimalkan untuk memenuhi kebutuhan pengguna dengan lebih baik [6].

Penelitian yang dilakukan oleh Bambang Dani Setiawan, Reza Avrizal, dan Siti Suaedah (2023), yang berjudul *Perancangan Sistem Aplikasi Gym Member Berbasis Android*, meneliti pembuatan aplikasi berbasis Android menggunakan *React Native* dan *MySQL*. Aplikasi ini bertujuan untuk mempermudah penyimpanan data member dan meningkatkan efisiensi dalam pencatatan aktivitas member gym. Dengan aplikasi ini, proses pendaftaran, transaksi, serta pengelolaan jadwal kelas dapat dilakukan lebih cepat dan akurat dibandingkan metode manual [7].

Penelitian yang dilakukan oleh Aprijal Muhaemin, *Perancangan Sistem Informasi Membership Fitness pada FR Fitness Cileungsi Berbasis Java NetBeans*, 2023, membahas pengembangan sistem informasi keanggotaan menggunakan *Java NetBeans* dan *MySQL*. Aplikasi ini bertujuan untuk mempermudah manajemen data member, seperti pendaftaran, pelaporan, dan transaksi, yang sebelumnya dilakukan secara manual, sehingga meminimalkan kesalahan data. Sistem ini dirancang untuk menggantikan metode manual yang sering menyebabkan kesalahan dan mempercepat waktu transaksi [8].

Penelitian lain yang dilakukan oleh Yuri Yuliani, Yuri Rahayu, Susilawati, dan Robi Aziz Zuama, *Perancangan Sistem Informasi Pengolahan Data Member pada Rai Fitness Sukabumi*, 2021, berfokus pada pembuatan sistem pengelolaan data member, termasuk pendaftaran dan

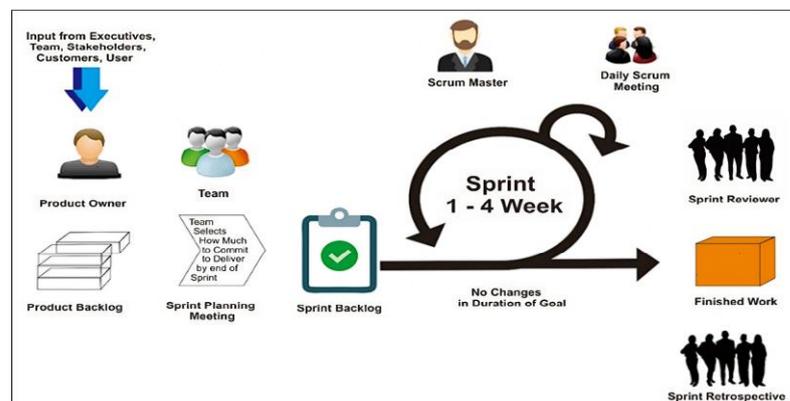
perpanjangan secara otomatis. Aplikasi ini dirancang untuk mencatat data member dengan lebih rapi dan efisien dibandingkan metode manual, sehingga mempercepat proses pembuatan laporan [9].

Angga Arindra Shontja, Laily Nur Hamidah, Muhammad Hasan, Melany Mustika Dewi, dan Yuli Astuti, *Penerapan Firebase Realtime Database pada Aplikasi Media Informasi dan Pendaftaran Training IT Berbasis Android*, (2022). Penelitian ini menjelaskan pengembangan aplikasi yang digunakan di *Jogja IT Training Center*. Aplikasi ini memanfaatkan *Firestore Realtime Database* sebagai platform untuk penyimpanan data pendaftaran secara *real-time*, bertujuan meningkatkan efisiensi proses pendaftaran dan penyebaran informasi. Hasilnya adalah aplikasi *Android* yang memudahkan manajemen pelatihan dan mempercepat pendaftaran peserta [10].

Penelitian ini berfokus pada pengembangan Aplikasi *Gym Member* yang bertujuan untuk memudahkan berbagai proses seperti pendaftaran, pembaruan profil, dan pengelolaan data keanggotaan secara *real-time*. Aplikasi ini dirancang khusus untuk *platform Android* dan memanfaatkan *Firestore Realtime Database* sebagai penyimpan datanya. Perbedaan utama dari penelitian sebelumnya terletak pada jenis database yang digunakan. Jika penelitian sebelumnya menggunakan database relasional, kali ini diterapkan database non-relasional dengan teknologi *Firestore Realtime Database*. Berbeda dari pendekatan sebelumnya, aplikasi ini menawarkan solusi yang lebih terintegrasi, di mana proses pendaftaran dapat diotomatisasi sepenuhnya. Selain itu, aplikasi ini memungkinkan sinkronisasi data secara cepat dan akurat di berbagai perangkat secara *real-time*, sehingga meningkatkan efisiensi pengelolaan data anggota gym.

### 3. Metodologi

Metodologi penelitian yang digunakan dalam penelitian ini menggunakan *Agile software development* dengan metode yang digunakan adalah metode *scrum* [11]. Metode *Scrum* telah banyak dimanfaatkan oleh para peneliti dalam proses pengembangan aplikasi, seperti halnya penerapan metode *Scrum* pada *Pengembangan Aplikasi Tiga-Tingkat di Aplikasi Presensi Karyawan Glints Academy* [12]. Penelitian ini menggunakan metode *Scrum* dengan tujuan untuk menghasilkan pengembangan aplikasi yang cepat, mampu beradaptasi dengan perubahan, serta sesuai dengan target yang diinginkan [13]. Metode pengembangan sistem menggunakan *Agile Scrum* ditunjukkan pada gambar 1.



Gambar 1. *Framework Agile Scrum* [14]

#### 3.1 Tahapan Pengembangan Sistem

Tahapan-tahapan yang dilaksanakan dalam metode *Agile Scrum* adalah sebagai berikut:

- 1) *Perencanaan*: Pada tahap perencanaan, dilakukannya analisis kebutuhan terhadap aplikasi yang akan dikembangkan, yaitu aplikasi *Gym Member* yang bertujuan untuk memudahkan proses pendaftaran, pembaruan profil, dan pengelolaan data keanggotaan. User stories atau kebutuhan fungsional sistem diidentifikasi dan disusun dalam *backlog produk*.
- 2) *Sprint Planning*: Dilakukan untuk menetapkan tujuan dan sasaran pekerjaan dalam satu periode sprint. Sprint berlangsung selama 2-4 minggu, dengan fokus pada penyelesaian fitur

tertentu dari aplikasi, seperti modul pendaftaran anggota pada sprint pertama. User stories yang ada dalam backlog dipilih dan dipecah menjadi tugas-tugas yang lebih kecil, kemudian diselesaikan secara bertahap selama setiap sprint.

- 3) *Pengembangan fitur*: Pada tahap ini, pengembangan fitur aplikasi dimulai berdasarkan tugas-tugas yang telah ditetapkan dalam sprint planning. Fokus diarahkan pada pencapaian target pekerjaan sesuai batas waktu sprint, termasuk mengimplementasikan *Firestore Realtime Database* untuk memastikan data tersinkronisasi dengan baik dan akurat.
- 4) *Review dan Retrospektif*: Setelah menyelesaikan setiap sprint, sebagai pengembang melakukan evaluasi terhadap fitur yang telah diselesaikan. Evaluasi ini mencakup peninjauan terhadap kualitas implementasi, kesesuaian dengan kebutuhan awal, serta identifikasi kendala yang dihadapi. Hasil dari evaluasi ini digunakan sebagai acuan untuk perbaikan pada sprint berikutnya, sehingga proses pengembangan tetap berjalan secara efisien dan optimal.
- 5) *Uji Coba dan Pemeliharaan*: Tahap uji coba dilakukan untuk memastikan bahwa sistem beroperasi dengan optimal dan sesuai dengan kebutuhan pengguna. Selama tahap pemeliharaan, dilakukannya perbaikan dan penyesuaian sesuai dengan masukan yang didapat dari pengguna dan dari hasil menjalankan aplikasi secara mandiri.

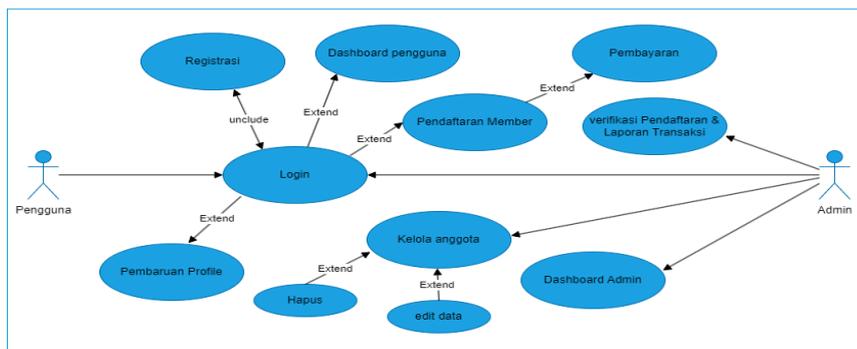
### 3.2 Perancangan Sistem

*Unified Modeling Language* (UML) adalah sebuah bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai media penulisan cetak biru (*blueprints*) perangkat lunak. UML bisa digunakan untuk visualisasi, spesifikasi, konstruksi, dan dokumentasi beberapa bagian-bagian dari sistem yang ada dalam perangkat lunak [15]. Diagram UML yang diterapkan dalam penelitian ini adalah:

- 1) *Use Case*: Digunakan untuk memvisualisasikan interaksi antara pengguna dan sistem serta fungsi yang ada di dalamnya.
- 2) *Sequence Diagram*: Memperlihatkan interaksi dinamis antara objek-objek dalam skenario tertentu.
- 3) *Class Diagram*: Menggambarkan struktur statis sistem, termasuk kelas-kelas, atribut, serta hubungan di antara mereka.

#### 3.2.1. Use Case Diagram

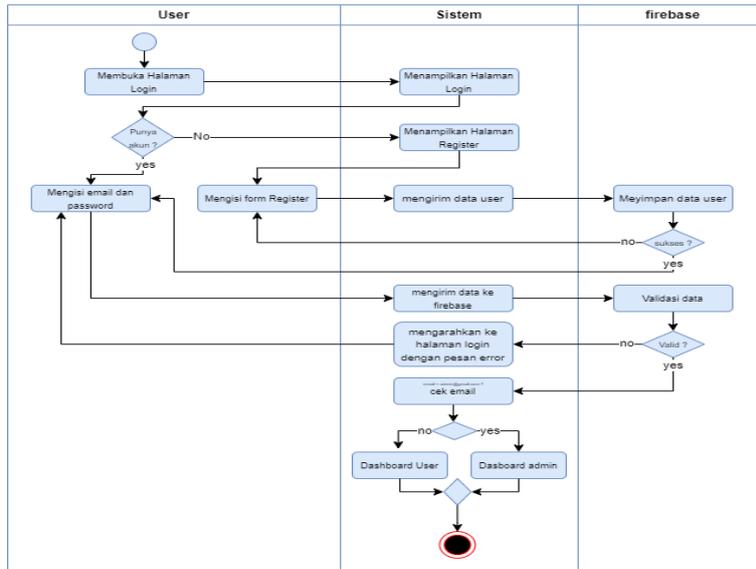
Sistem ini melibatkan dua tipe pengguna, yaitu Pengguna dan Admin, yang digambarkan dalam Gambar 2.



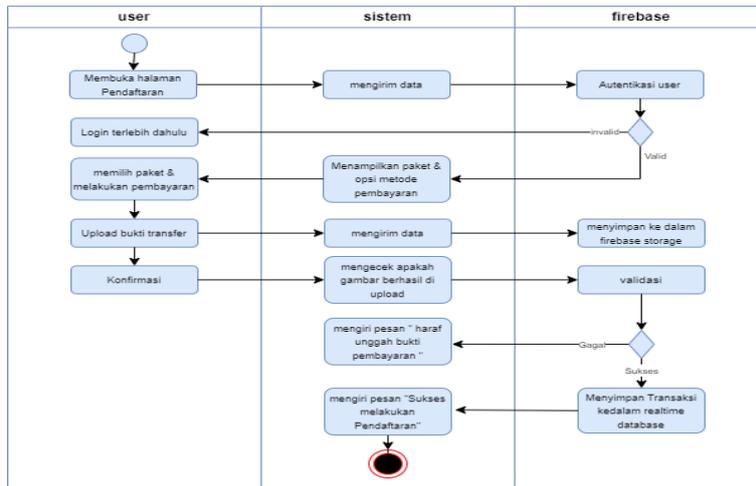
Gambar 2. Use Case Diagram

#### 3.2.2. Activity Diagram

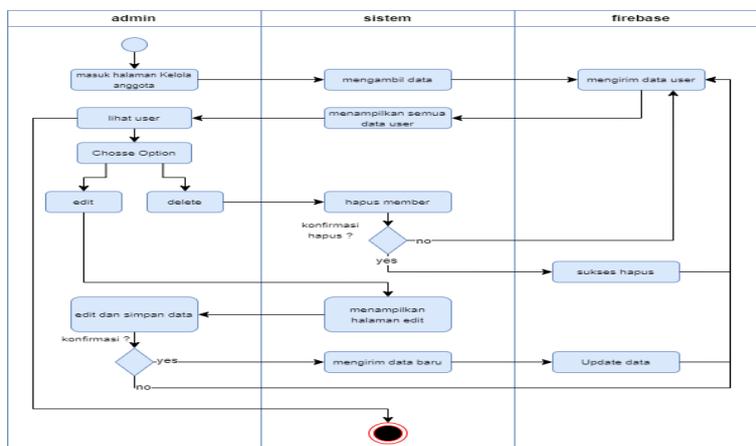
*Activity diagram* adalah aktivitas yang dilakukan oleh sistem. Terdapat tiga activity diagram utama yang menjadi fitur utama dalam aplikasi ini, yaitu: Login & Registrasi, Pendaftaran, serta Pengelolaan Anggota.



Gambar 3. Activity Diagram Login & Registrasi



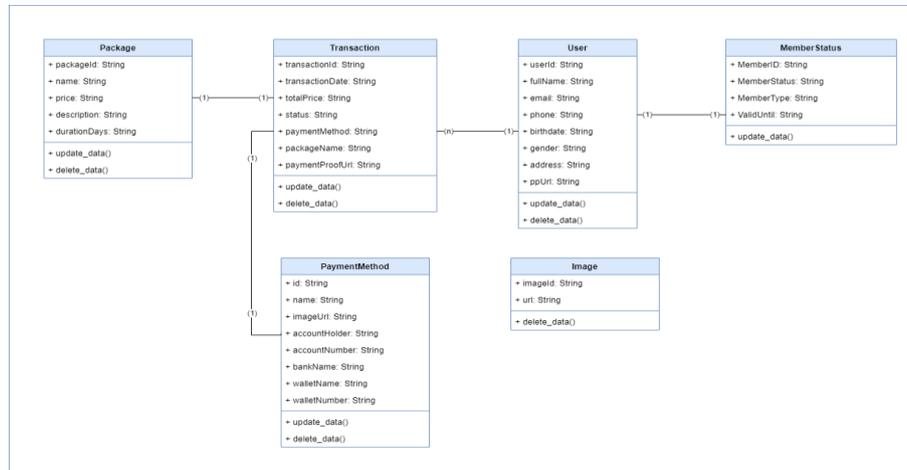
Gambar 4. Activity Diagram Pendaftaran



Gambar 5. Activity Diagram Kelola anggota

### 3.2.3. Class Diagram

Diagram kelas menggambarkan struktur objek dalam sistem, termasuk atribut dan metode dari setiap kelas, serta hubungan antar kelas tersebut. Aplikasi *Gym Member* memiliki 7 kelas.



Gambar 6. Class Diagram

### 3.3 Implementasi Sistem

Tahap implementasi adalah proses penerapan atau pengoperasian sistem yang telah dirancang dan dikembangkan. Tujuannya memastikan sistem berjalan sesuai dengan kebutuhan. Pada implementasi ini, sistem dibagi ke dalam beberapa komponen utama:

- 1) Implementasi *Realtime Database*. *Firebase* digunakan untuk mengelola data secara *real-time*, memastikan sinkronisasi data antara aplikasi klien dan server berjalan lancar.
- 2) Autentikasi Pengguna. Menggunakan *Firebase Authentication*, pengguna dapat mendaftar dan masuk ke dalam sistem dengan autentikasi email dan kata sandi.
- 3) Pengujian Fungsionalitas Sistem. Pengujian dilakukan untuk memastikan semua fitur, seperti registrasi anggota, login, dan sinkronisasi data, berjalan sesuai dengan spesifikasi. Uji coba mencakup pengujian di berbagai perangkat untuk memastikan kompatibilitas.

#### 3.3.1. Penerapan Firebase Realtime Database

*Firebase* merupakan suatu layanan dari *Google* yang digunakan pengembang aplikasi dalam mengembangkan aplikasi secara mudah dan aman [16]. Untuk mengintegrasikan *Firebase Realtime Database* ke dalam aplikasi *Android*, langkah pertama adalah menambahkan dependensi *Firebase* ke dalam *file Gradle*. Pada bagian ini, perlu mendeklarasikan modul *Firebase* yang dibutuhkan agar aplikasi dapat memanfaatkan berbagai layanan *Firebase*. Gambar.2 memperlihatkan deklarasi semua modul *Firebase* yang dibutuhkan agar aplikasi dapat menggunakan *firebase* pada sistem. Langkah deklarasi ini penting untuk memastikan aplikasi dapat mengimplementasikan database dengan optimal.

```

dependencies {
    implementation(platform("com.google.firebase:firebase-bom:33.1.2"))
    implementation("com.google.firebase:firebase-analytics")
    implementation("com.google.firebase:firebase-auth-ktx:22.0.0")
    implementation("com.google.firebase:firebase-auth")
    implementation("com.google.firebase:firebase-firestore")
    implementation("com.google.firebase:firebase-database:20.0.5")
    implementation("com.google.firebase:firebase-storage:20.0.1")
}
  
```

Gambar 7. Deklarasi Untuk Menerapkan Firebase

Setelah *dependensi Firebase* ditambahkan dan *Firebase* berhasil diinisialisasi, langkah selanjutnya adalah mengimplementasikan kode untuk mengelola interaksi dengan *Firebase Realtime Database*. Salah satu implementasi penting adalah proses Registrasi yang disimpan

dalam *Firestore Realtime Database* secara *real-time*. Berikut adalah contoh kode untuk mengunggah data Registrasi ke *Firestore*:

```
private fun registerUser(fullName: String, email: String, password: String, phone: String, gender: String) {
    auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                val userId = auth.currentUser?.uid
                val database = FirebaseDatabase.getInstance()
                val usersRef = database.getReference("Users")

                val user = mapOf(
                    "fullName" to fullName,
                    "email" to email,
                    "phone" to phone,
                    "gender" to gender
                )

                userId?.let { it: String
                    usersRef.child(it).setValue(user)
                    .addOnSuccessListener { it: Void
                        Toast.makeText(context, this, text: "Registration successful!", Toast.LENGTH_SHORT).show()
                        startActivity(Intent(context, LoginActivity::class.java))
                        finish()
                    }
                    .addOnFailureListener { it: Exception
                        Toast.makeText(context, this, text: "Failed to save user data: ${it.message}", Toast.LENGTH_SHORT).show()
                    }
                }
            } else {
                Toast.makeText(context, this, text: "Registration failed: ${task.exception?.message}", Toast.LENGTH_SHORT).show()
            }
        }
}
```

Gambar 8. mengunggah data Registrasi ke *Firestore*

Kode digunakan untuk *registrasi pengguna ke Firestore* dengan email dan password melalui *Firestore Authentication*. Data seperti nama, email, nomor telepon, dan jenis kelamin disimpan di *Firestore Realtime Database*. Setelah akun berhasil dibuat, *userId* digunakan sebagai kunci unik di node "Users". Jika berhasil, pengguna diarahkan ke halaman login, dan jika gagal, ditampilkan pesan error. Setiap perubahan data di *Realtime Database* akan otomatis tersinkronisasi secara *real-time* dengan semua klien yang terhubung.

### 3.3.2. Autentikasi Pengguna dengan *Firestore Authentication*

*Firestore Authentication* mendukung sistem autentikasi yang aman sekaligus mempercepat proses login dan pendaftaran akun pengguna [17]. *Firestore Authentication* memungkinkan autentikasi secara aman menggunakan berbagai metode, seperti email dan password, nomor telepon, atau akun sosial media. Dalam aplikasi ini, metode email dan password digunakan karena sederhana dan sesuai dengan kebutuhan pengelolaan anggota aplikasi *Gym Member*.

### 3.4 Pengujian Sistem

Pengujian sistem dalam metodologi *Scrum* dilakukan secara iteratif pada setiap sprint untuk memastikan kualitas setiap fitur yang dikembangkan. Proses pengujian mencakup:

- 1) *Pengujian Unit*: Memastikan setiap fungsi dalam aplikasi, seperti fitur registrasi dan login, berjalan dengan benar. Meskipun hasilnya tidak disajikan dalam penelitian, pengujian ini memastikan bahwa setiap fungsi berjalan sesuai spesifikasi. Misalnya, Pengujian registrasi memastikan bahwa data pengguna berhasil disimpan ke *Firestore Authentication* dan *Firestore Realtime Database* dengan benar. Kemudian ada Pengujian fitur login memastikan bahwa hanya pengguna dengan informasi akun yang valid dapat mengakses sistem. Proses verifikasi ini diuji dengan berbagai skenario, sehingga pengguna dengan data login yang salah menerima pesan error yang sesuai, sementara pengguna yang valid berhasil masuk. Pengujian unit ini menjamin bahwa fitur registrasi dan login berfungsi sesuai kebutuhan, mendukung keamanan dan stabilitas sistem.
- 2) *Pengujian Integrasi*: Pengujian integrasi bertujuan untuk memeriksa interaksi antar komponen dalam sistem, memastikan bahwa setiap komponen berfungsi secara sinkron dan tanpa kesalahan. Dalam penelitian ini, pengujian integrasi dilakukan untuk memverifikasi bahwa aplikasi *Gym Member* dapat berinteraksi dengan *Firestore Realtime Database* dengan baik, khususnya dalam mengelola data anggota secara *real-time*.
- 3) *Pengujian Fungsional*: Menguji fitur utama aplikasi berdasarkan kebutuhan pengguna. Pengujian fungsional dalam proyek ini, seperti yang terlihat pada skenario pengujian *black-*

box seperti Pendaftaran Membership, Pembaruan Status Membership, dan Pengelolaan Data Anggota.

Pengujian sistem pada aplikasi *Gym Member* berjalan sukses. Setiap fitur utama, seperti registrasi dan login, berfungsi sesuai spesifikasi. Integrasi antar komponen dengan *Firestore Realtime Database* berjalan lancar, dan pengujian fungsional memastikan aplikasi memenuhi kebutuhan pengguna dengan baik.

#### 4. Hasil dan Pembahasan

##### 4.1 Product Backlog

*Product backlog* adalah langkah awal pada pengembangan sistem yang menggunakan metode *Scrum*. Pada tahap ini, akan dihasilkan berbagai elemen seperti aktor, proses sistem, aktivitas sistem, dan struktur sistem. *Product backlog* untuk pengembangan sistem dalam penelitian ini ditampilkan pada Tabel.

Tabel 1. *Product backlog*

| No | Item                                  | Prioritas |
|----|---------------------------------------|-----------|
| 1  | Perancangan Sistem dan Database       | tinggi    |
| 2  | Login dan Register                    | sedang    |
| 3  | Halaman Pendaftaran                   | tinggi    |
| 4  | Halaman kelola data Keanggotaan       | tinggi    |
| 5  | Pembaruan otomatis status membership  | tinggi    |
| 5  | Integrasi Firestore Realtime Database | tinggi    |
| 6  | Halaman Dashboard Pengguna            | rendah    |
| 7  | Halaman Dashboard Admin               | rendah    |

Tujuan dari *product backlog* adalah untuk membangun sistem dan struktur data yang solid serta memungkinkan sinkronisasi *real-time* untuk data keanggotaan. Tujuan dari *product backlog* nomor 2 hingga 7 adalah untuk mengembangkan fitur login, registrasi, pengelolaan keanggotaan, serta sinkronisasi dengan *Firestore Realtime Database* untuk meningkatkan efisiensi pendaftaran dan pengelolaan data member pada aplikasi *Gym Member* berbasis Android.

##### 4.2 Pengujian Sistem

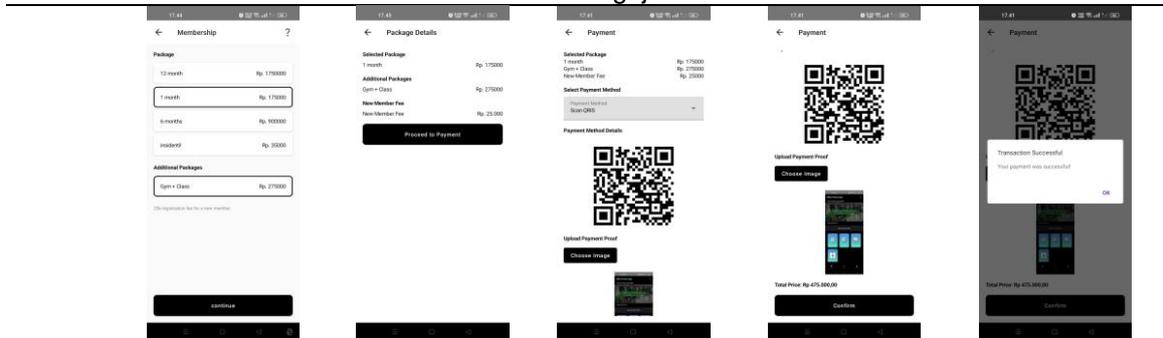
Tahap pengujian adalah komponen penting dalam pengembangan perangkat lunak atau implementasi proyek untuk memastikan sistem berfungsi sesuai spesifikasi dan dapat diandalkan. Pada Pengujian kali ini saya menggunakan metode *black-box*, yang bertujuan untuk mengidentifikasi kesalahan sistem agar sistem dapat dijalankan secara optimal [18].

Tabel 2. Hasil Pengujian Fitur Pendaftaran Membership

|                   |   |            |
|-------------------|---|------------|
| Nama Kasus Uji    | Pendaftaran member  |            |
| Deskripsi         | User melakukan Pendaftaran member                                     |            |
| Kondisi awal      | User sudah berhasil login<br>User berada dihalaman dashboard pengguna |            |
| Tanggal pengujian | 20 September 2024   |            |
|                   | Skenario  |            |
|                   | 1) Klik tombol "Aktivasi Membership"                                  |            |
|                   | 2) Klik paket yang akan dibeli  |            |
|                   | 3) Klik tombol "Continue"   |            |
|                   | 4) Klik tombol "Proceed to Payment"                                   |            |
|                   | 5) Pilih metode pembayaran  |            |
|                   | 6) Klik tombol "Choose Image"   |            |
|                   | 7) Pilih gambar (bukti pembayaran)                                    |            |
|                   | 8) Klik tombol "Confirm"  |            |
|                   | 9) Muncul pop up jika transaksi sukses                                |            |
| Yang diharapkan   | Pengamatan  | Kesimpulan |
| Data transaksi    | berhasil Pendaftaran atau aktivasi                                    | Sesuai.    |

disimpan kedalam Realtime member berhasil. Data Database. berhasil disimpan kedalam Realtime Database Firebase.

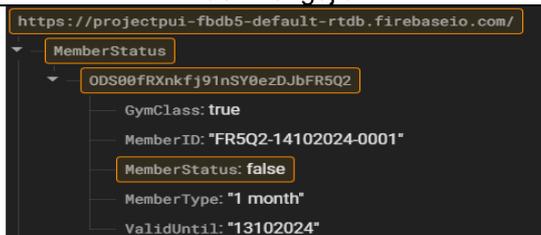
Hasil Pengujian



Tabel 3. Hasil Pengujian Fitur Pembaruan Status Membership

|                                     |  |
|-------------------------------------|--|
| Nama Kasus Uji                      | Pembaruan status membership  |
| Deskripsi                           | Uji coba pembaruan status membership otomatis berdasarkan masa berlaku paket dengan waktu sekarang secara realtime. Dengan catatan user mengakses aplikasi dan sistem akan mengecek apakah masa berlaku sudah habis ? jika iya sistem akan mengubah status membership menjadi false. |
| Kondisi awal                        | User sudah melakukan pendaftaran<br>User melakukan pendaftaran dengan membeli paket 1 bulan<br>Masa berlaku tinggal 1 hari (contoh)  |
| Tanggal pengujian                   | 3 Oktober 2024   |
|                                     | Skenario   |
|                                     | 1) Masa berlaku paket tinggal satu hari<br>2) Sistem melakukan perubahan status otomatis jika masa berlaku sudah habis   |
| Yang diharapkan                     | Pengamatan   |
| Pembaruan status otomatis berhasil. | Pembaruan berhasil.  |
|                                     | Kesimpulan   |
|                                     | Sesuai.  |

Hasil Pengujian



Tabel 4. Hasil Pengujian Fitur Kelola Data Anggota

|                   |  |
|-------------------|--|
| Nama Kasus Uji    | Kelola data anggota  |
| Deskripsi         | Pengujian untuk memastikan fitur kelola anggota berfungsi dengan baik, mengedit, menghapus, dan memeriksa status keanggotaan.  |
| Kondisi awal      | Sistem telah memiliki beberapa anggota terdaftar dengan data seperti nama, email, ID member, dan tipe keanggotaan. Pengguna memiliki akses sebagai admin untuk mengelola data anggota. |
| Tanggal pengujian | 4 Oktober 2024   |
|                   | Skenario   |

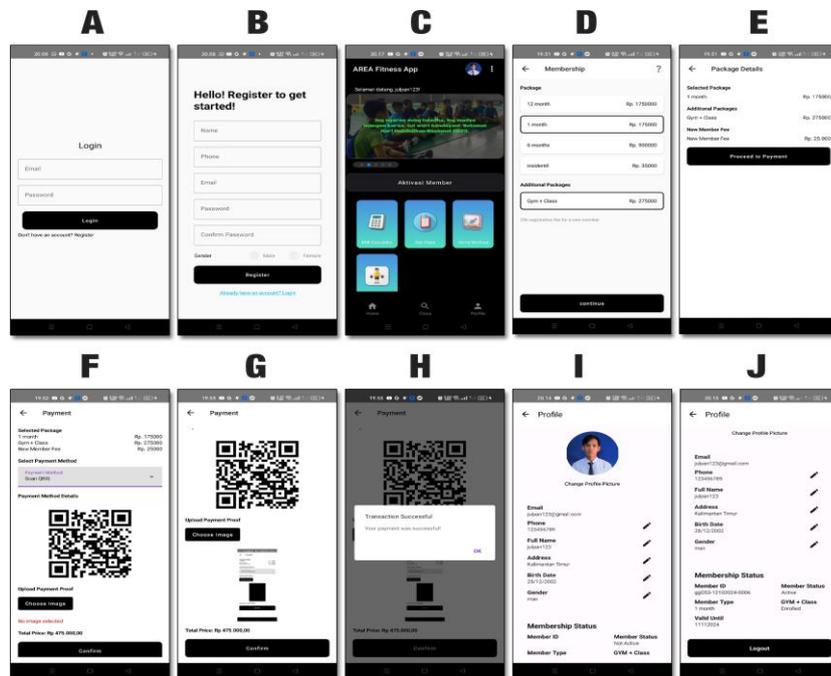
- 1) Admin mengedit informasi anggota dan menyimpan perubahan
- 2) Admin menghapus salah satu anggota yang tidak aktif.
- 3) Sistem otomatis menandai anggota baru sebagai aktif jika transaksi sukses.

| Yang diharapkan   | Pengamatan   | Kesimpulan   |
|---|--|--|
| <ol style="list-style-type: none"> <li>1) Informasi anggota diperbarui dengan sukses dan ditampilkan di tabel anggota.</li> <li>2) Anggota yang dihapus tidak lagi terlihat dalam tabel anggota.</li> <li>3) Anggota baru berhasil ditandai sebagai aktif jika transaksi sukses dan divalidasi oleh admin.</li> </ol> | Data anggota berhasil diperbarui sesuai dengan perubahan yang dilakukan. | Semua skenario pengelolaan anggota berjalan sesuai harapan. Sistem kelola anggota berfungsi dengan baik, termasuk fitur edit, hapus, dan tanda status aktif. |

Hasil Pengujian



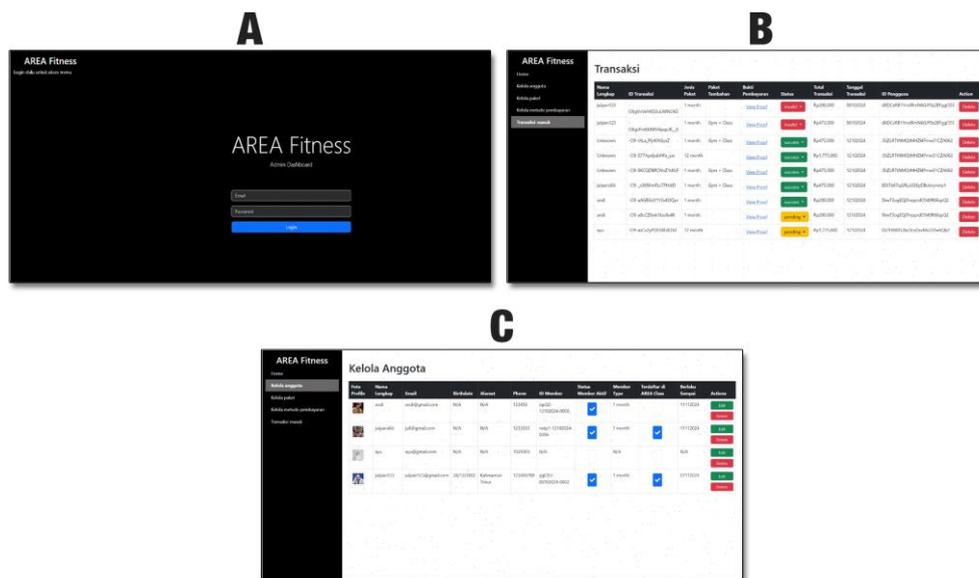
4.3 Tampilan Antarmuka Aplikasi



Gambar 9. Tampilan Antarmuka Pengguna

- 1) *Login* : Gambar A berisi login, jika login berhasil akan diarahkan ke halaman Dashboard Pengguna.
- 2) *Register* : Gambar B berisi register. Terdapat nama, nomer handphone, email, jenis kelamin dan password.

- 3) *Dashboar Pengguna* : Pada gambar C berisi halaman dashboard pengguna.
- 4) *Pilih paket* : Gambar D berisi halaman pilih paket. Setelah pengguna memilih paket dan klik tombol "Continue" akan mengarahkan ke halaman detail paket.
- 5) *Detail Paket* : Gambar E berisi halaman untuk menampilkan paket yang dipilih sebelumnya. Ketika pengguna klik tombol "proceed to paymet" akan mengarahkan ke halaman pembayaran.
- 6) *Lanjut ke halaman pembayaran* : Gambar F,G,H adalah halaman untuk melakukan proses pembayaran. Pada halaman ini pengguna harus memilih metode pembayaran dan melakukan upload gambar sebagai bukti pembayaran.
- 7) *Profile* : Gambar I dan J adalah halaman profile. Pada halaman ini digunakan untuk menampilkan data pengguna dan status membership.



Gambar 10. Tampilan Antarmuka Admin

- 1) *Login Admin* : Pada gambar A adalah halaman untuk login Admin yang berisi email dan password yang sudah diberikan dan jika login berhasil akan mengarahkan ke halaman kelola anggota.
- 2) *Kelola Transaksi Masuk* : Pada gambar B adalah halaman untuk mengelola transaksi yang masuk dari pendaftaran member. Admin dapat mengubah status transaksi dari pending menjadi success atau invalid. Jika status diubah menjadi success maka status anggota yang melakukan pendaftaran akan otomatis menjadi aktif.
- 3) *Kelola anggota* : Pada gambar C adalah halaman untuk mengelola data anggota.

#### 4.4 Pembahasan

Pada penelitian ini, permasalahan utama yang dihadapi adalah pengelolaan data keanggotaan gym yang masih dilakukan secara manual, sehingga proses pendaftaran member dan pengelolaan data lainnya menjadi tidak efisien. Penelitian ini mengembangkan aplikasi *Gym Member berbasis Android* yang dilengkapi dengan *Firestore Realtime Database* dan metode *Agile Scrum* untuk meningkatkan efisiensi dalam pengelolaan data keanggotaan gym secara *real-time*.

Penelitian ini memiliki beberapa kesamaan dengan penelitian sebelumnya yang juga mengembangkan sistem berbasis aplikasi untuk memudahkan manajemen keanggotaan. Contohnya, penelitian yang berjudul *Perancangan Sistem Aplikasi Gym Member Berbasis Android* yang mengembangkan aplikasi gym berbasis *Android* menggunakan *MySQL* untuk mempermudah pencatatan aktivitas member [7]. Meskipun memiliki tujuan yang sama, penelitian ini menawarkan kelebihan dengan penerapan *Firestore Realtime Database* yang memungkinkan sinkronisasi data secara *real-time*, sehingga memberikan solusi yang lebih

efisien dibandingkan dengan metode database tradisional yang digunakan dalam penelitian sebelumnya.

Penelitian ini menggunakan metode *Agile Scrum*, yang memberikan fleksibilitas dalam proses pengembangan aplikasi melalui sprint yang terorganisir. Pendekatan ini memungkinkan pengembangan yang lebih responsif terhadap perubahan kebutuhan pengguna, berbeda dengan metode lain yang lebih kaku, seperti *Grounded Research* yang diterapkan pada penelitian sebelumnya. *Agile Scrum* sangat sesuai untuk proyek seperti aplikasi *Gym Member*, di mana kebutuhan pengguna bisa berubah selama proses pengembangan, sehingga aplikasi dapat terus disempurnakan dalam tiap sprint berdasarkan feedback pengguna.

## 5. Simpulan

Kesimpulan dari penelitian ini adalah bahwa penerapan Firebase Realtime Database dalam pengembangan aplikasi Gym Member berbasis Android telah berhasil meningkatkan efisiensi proses pendaftaran, pembaruan profil, dan pengelolaan data keanggotaan secara real-time. Penggunaan metode Agile Scrum dalam pengembangan memungkinkan fleksibilitas dan penyesuaian sesuai kebutuhan pengguna secara berkelanjutan. Aplikasi ini mampu menyinkronkan data secara cepat dan akurat, serta mengotomatisasi beberapa proses penting seperti pembaruan status keanggotaan. Dengan demikian, aplikasi ini mampu mengatasi kendala pengelolaan keanggotaan yang sebelumnya dilakukan secara manual dan rentan terhadap kesalahan atau kehilangan data, sehingga memberikan solusi yang lebih efisien dan modern untuk operasional pusat kebugaran.

## Daftar Referensi

- [1] F. H. Putra, A. Aditya, and S. Sakaria, "Sistem Informasi Membership Dan Pengaturan Jadwal Untuk Mempermudah Administrasi Pada Alto Myshouse Berbasis Web," *J-Intech*, vol. 10, no. 1, pp.55–63, 2022, doi: 10.32664/j-intech.v10i1.679.
- [2] Khairun Nisa Meiah Ngafidin, A. Arista, and R. Nisa Sofia Amriza, "Implementasi Firebase Realtime Database pada Aplikasi FeedbackMe sebagai Penghubung Guru dan Orang Tua," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp.327–334, 2021, doi: 10.29207/resti.v5i2.2909.
- [3] I. Humaini, "Jurnal Pengabdian Masyarakat dan aplikasi Teknologi (Adipati) Aplikasi Mobile Pemesanan Jasa Barbershop Menggunakan Firebase Realtime Database," *J. ADIPATI*, vol. 01, no. 02, pp.60–67, 2022, [Online]. Available: <https://ejurnal.itats.ac.id/adipati/>
- [4] A. S. W. Jelantik, P. T. H. Permana, and N. M. Estiyanti, "Analisis Dan Perancangan Sistem Point of Sales Menggunakan Metode Agile Development Pada Toko Eka Putra Sukawati," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 2, p. 185, 2021, doi: 10.35889/jutisi.v10i2.660.
- [5] B. Susilo and A. Azimah, "Penerapan Metode Agile Scrum Pada Perancangan Sistem Informasi Pelaporan Keuangan BUMDesa," *JUTISI J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 18, no. 2, pp.495–505, 2023.
- [6] M. Lestari, P. Mendrofa, and M. H. Simbolon, "Penerapan Metode Prototype pada Perancangan Aplikasi Pencatatan Penjualan Sales Berbasis Web," vol. 3, no. 2, pp.10–15, 2024, doi: 10.58918/lofian.v4i1.273.
- [7] B. D. Setiawan, R. Avrizal, and S. Suaedah, "Perancangan Sistem Aplikasi Gym Member Berbasis Android," *JRKT (Jurnal Rekayasa Komputasi Ter.)*, vol. 3, no. 02, pp.88–95, 2023, doi: 10.30998/jrkt.v3i02.8600.
- [8] A. Muhaemin, "Perancangan Sistem Informasi Membership Fitness pada FR Fitness Cileungsi Berbasis Java Netbeans," *J. Ris. dan Apl. Mhs. Inform.*, vol. 4, no. 01, pp.32–39, 2023, doi: 10.30998/jrami.v4i01.4680.
- [9] Y. Yuliani, Y. Rahayu, S. Susilawati, and R. A. Zuama, "Perancangan Sistem Informasi Pengolahan Data Member Pada Rai Fitness Sukabumi," *Informatics Comput. Eng. J.*, vol. 1, no. 2, pp.45–53, 2021, doi: 10.31294/icej.v1i2.475.
- [10] A. A. Shonta, L. N. Hamidah, M. Hasan, M. M. Dewi, Y. Astuti, and I. R. Wulandari, "Penerapan Firebase Realtime Database Pada Aplikasi Media Informasi dan Pendaftaran Training IT Berbasis Android," *J. Media Inform. Budidarma*, vol. 6, no. 3, p. 1517, 2022, doi: 10.30865/mib.v6i3.4040.
- [11] K. Schwaber and J. Sutherland, "Scrum Guide V7," *Agil. Metrics Agil. Heal. Metrics*

- Predict.*, no. November, pp.133–152, 2020.
- [12] I. Tahyudin and Zidni Iman Sholihati, "Pengembangan Aplikasi Tiga-Tingkat Menggunakan Metode Scrum pada Aplikasi Presensi Karyawan Glints Academy," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 1, pp.169–176, 2022, doi: 10.29207/resti.v6i1.3793.
- [13] D. J. K. Putra and P. F. Tanaem, "Perancangan Aplikasi Pembukuan Menggunakan Metode Agile Scrum," *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 3, pp.509–521, 2022, doi: 10.28932/jutisi.v8i3.5060.
- [14] D. W. A. Nugroho, "Rancang Bangun Sistem Informasi Gelanggang Olahraga berbasis Web dengan Metode Scrum," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 8, no. 4, pp.1733–1749, 2021, doi: 10.35957/jatisi.v8i4.1132.
- [15] R. Abdillah, "Pemodelan Uml Untuk Sistem Informasi Persewaan Alat Pesta," *J. Fasilkom*, vol. 11, no. 2, pp. 79–86, 2021, doi: 10.37859/jf.v11i2.2673.
- [16] R. K. Dinata and O. Hamzah, "170-Article Text-727-1-10-20211220," vol. 2, no. 2, pp.2–8, 2021.
- [17] P. Aplikasi, G. Berbasis, and C. Computing, "Implementasi Platform As A Service (PAAS) Pada Aplikasi Getfix Berbasis Cloud Computing," *J. Sains dan Inform.*, vol. 8, no. 2, pp.86–95, 2022, doi: 10.22216/jsi.v8i2.1653.
- [18] I. Ismail and J. Efendi, "Black-Box Testing : Analisis Kualitas Aplikasi Source Code Bank Programming," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 4, no. 2, p. 1, 2020, doi: 10.35870/jtik.v5i1.148.