

I-Read: Aplikasi Agile untuk Pencarian Lokasi Buku Lintas Perpustakaan di Wilayah Makassar

Andi Mahdaniar Patoppoi. Z^{1*}, Alia Amini Azhar², Sayla Fardhillah³

^{1,2,3} Universitas Negeri Makassar, Indonesia

Corresponding e-mail : andimahdaniar27@gmail.com

INFO ARTIKEL

Kata Kunci:

Agile
Buku
Lokasi
Pencarian
Perpustakaan

Riwayat Artikel

Menerima: Januari 02,
2024
Revisi: Maret 10, 2024
Diterima: April 28, 2024

ABSTRAK

Perkembangan teknologi informasi telah mendorong transformasi layanan perpustakaan menuju sistem yang lebih modern dan efisien. Penelitian ini bertujuan untuk merancang aplikasi i-Read berbasis metode Agile guna mempermudah pengguna dalam mencari dan mengakses informasi buku di perpustakaan-perpustakaan di Makassar. Metode Agile dipilih karena fleksibilitasnya dalam menyesuaikan kebutuhan pengguna melalui iterasi yang cepat. Aplikasi ini dilengkapi dengan fitur pencarian buku berdasarkan judul, penulis, kategori, dan lokasi penyimpanan, serta informasi ketersediaan buku secara *real-time*. Proses pengembangan meliputi tahapan perencanaan, analisis kebutuhan, perancangan, implementasi, dan pengujian. Aplikasi diuji menggunakan metode Black Box dan White Box. Hasilnya menunjukkan bahwa i-Read dapat meningkatkan efisiensi layanan pencarian buku serta memberikan pengalaman pengguna yang lebih interaktif dan responsif. Penelitian ini diharapkan menjadi solusi inovatif dalam digitalisasi layanan perpustakaan dan acuan bagi pengembangan sistem serupa di instansi lainnya.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



To cite this article : Author. (20xx). Title. Information Technology Education Journal, X(X), XX-XX.
Doi. xxx

PENDAHULUAN

Perpustakaan merupakan ruangan atau bangunan yang menyimpan koleksi buku yang tersusun rapi agar mudah diakses saat dibutuhkan. Pengelolaan koleksi secara manual dapat menimbulkan masalah jika tidak ditata dengan baik. Selain sebagai tempat membaca di waktu luang, perpustakaan juga berperan sebagai sarana, alat, dan sumber belajar bagi siswa [1]. Buku merupakan sarana utama dalam memperoleh pengetahuan dan informasi, khususnya bagi kalangan pelajar dan mahasiswa.

Dalam proses pembelajaran, mereka kerap ditugaskan untuk mencari referensi tambahan maupun informasi pendukung melalui bahan bacaan tersebut [2]. Perguruan tinggi menyediakan layanan peminjaman buku melalui fasilitas perpustakaan. Namun demikian, tidak jarang buku yang dibutuhkan tidak tersedia atau jumlahnya terbatas. tidak tersedia atau stoknya terbatas, sehingga mahasiswa membuang waktu untuk mencari secara langsung dengan kemungkinan menemukan buku yang sesuai hanya sekitar 50% [3]. Kondisi ini mengakibatkan pengguna membuang waktu dan tenaga secara tidak efisien, terutama di wilayah perkotaan seperti Makassar, yang memiliki banyak institusi pendidikan dan perpustakaan tersebar di berbagai lokasi.

Salah satu tantangan utama yang dihadapi Indonesia dalam bidang pendidikan di era globalisasi adalah rendahnya kualitas sumber daya manusia. Upaya untuk meningkatkan kualitas

tersebut dapat dilakukan dengan mendorong peningkatan minat serta membiasakan kegiatan membaca. Dalam hal ini, perpustakaan diharapkan berperan sebagai pusat pengembangan minat baca, mengingat peran dan fungsinya yang strategis dalam mendorong peningkatan budaya literasi masyarakat [4]. Sehingga Sistem Informasi perpustakaan sangat penting keberadaannya.

Sejumlah penelitian sebelumnya telah mengembangkan sistem informasi perpustakaan dengan pendekatan yang beragam. Misalnya, Penelitian ini sebelumnya dilakukan oleh [5]. Perancangan sistem informasi perpustakaan di SMA Islam Al-Mizan menggunakan algoritma Levenshtein Distance bertujuan untuk mempermudah pencarian buku berdasarkan kemiripan kata. Sementara itu, Sutikman dan Winarsih mengembangkan aplikasi perpustakaan "MBAKU" yang difokuskan pada digitalisasi layanan peminjaman dan pendataan koleksi buku. Adapun pengembangan sistem perpustakaan di SMPN 11 Depok dilakukan dengan mengintegrasikan QR Code guna mempercepat proses peminjaman dan pengembalian buku [1].

Berbagai pengembangan sistem informasi perpustakaan telah dilakukan sebelumnya. Di SDN Cawang 12 Pagi, sistem manajemen buku daring dikembangkan untuk mempermudah pengelolaan koleksi [6]. Di RPTRA Lenteng Agung, aplikasi digital ramah anak dirancang untuk menyajikan koleksi secara menarik [7]. Penelitian lain mengembangkan sistem berbasis Java dengan teknologi barcode untuk meningkatkan efisiensi pengelolaan perpustakaan [8], serta implementasi di SMAN 6 Depok [9], dan SMKN 1 Lantuka [10]. Selain itu, aplikasi perpustakaan berbasis web juga dikembangkan menggunakan CodeIgniter di SMP Negeri 3 Pacet [11].

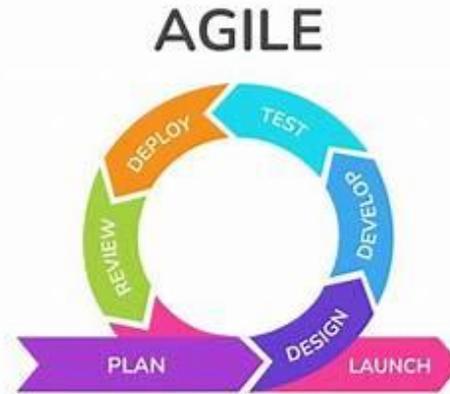
Hingga saat ini, belum terdapat penelitian yang secara khusus mengembangkan aplikasi pencarian lokasi buku lintas perpustakaan dengan fokus wilayah Makassar sebagaimana yang dilakukan dalam pengembangan i-Read. Penelitian sebelumnya umumnya masih terbatas pada digitalisasi sistem perpustakaan dalam lingkup satu institusi, tanpa menyediakan fitur pencarian lokasi buku antar perpustakaan maupun cakupan wilayah tertentu.

Melihat permasalahan tersebut, maka Pengembangan aplikasi i-Read menjadi relevan bagi masyarakat Makassar karena memudahkan pencarian buku lintas perpustakaan dengan sistem peta terintegrasi. i-Read tidak hanya menyediakan informasi, tetapi juga meningkatkan efisiensi pencarian, memperluas akses bacaan, dan memperkuat peran perpustakaan dalam pengembangan literasi di wilayah urban.

METODE

Pada penelitian ini digunakan metode Agile Developments. Metode Agile Development merupakan sekumpulan metodologi dalam pengembangan perangkat lunak yang berbasis pada pendekatan iteratif [12]. Artinya, pengembangan dilakukan secara bertahap dan berulang, memungkinkan penyempurnaan sistem secara berkelanjutan. Metode agile memungkinkan tim mengembangkan perangkat lunak secara cepat, adaptif, dan sesuai kebutuhan pengguna [13].

Metode Agile Development memungkinkan tim pengembang menetapkan prioritas fitur dan beradaptasi dengan perubahan selama proses pengembangan aplikasi [14]. Dalam praktiknya, tim bekerja dalam iterasi yang lebih pendek dan cepat, memungkinkan pengembangan aplikasi secara bertahap namun berkesinambungan. Tahapan-tahapan dalam metode Agile dapat dilihat pada Gambar 1.



Gambar 1. Tahapan – Tahapan Metode Agile

- 1) Perencanaan (*Plan*) : Identifikasi kebutuhan pengguna, tujuan aplikasi, dan ruang lingkup pengembangan melalui diskusi dengan calon pengguna, khususnya terkait pencarian buku dan lokasi perpustakaan di wilayah Makassar.
- 2) Perancangan (*Design*) : Perancangan antarmuka pengguna (UI) dan pengalaman pengguna (UX) serta struktur navigasi aplikasi yang intuitif dan sesuai kebutuhan masyarakat lokal.
- 3) Pengembangan (*Develop*): Implementasi fitur utama berdasarkan desain dan user story, seperti pencarian buku, pencarian perpustakaan, kategori buku, dan fitur rekomendasi secara bertahap.
- 4) Pengujian (*Test*) : Pelaksanaan uji unit, integrasi, dan antarmuka untuk memastikan semua fitur berfungsi sesuai kebutuhan pengguna.
- 5) Penerapan (*Deploy*) : Penyebaran versi awal aplikasi ke platform pengujian untuk evaluasi performa dan keandalan oleh tim pengembang dan stakeholder terbatas.
- 6) Tinjauan (*Review*) : Evaluasi hasil pengujian bersama tim dan pengguna, guna memperoleh masukan untuk perbaikan dan iterasi berikutnya.
- 7) Peluncuran (*Launch*) : Peluncuran resmi aplikasi versi stabil kepada publik di wilayah Makassar sebagai produk akhir yang siap digunakan secara efektif.

Subjek dan Objek Penelitian

Objek penelitian ini adalah aplikasi i-Read, yang dikembangkan untuk pencarian lokasi buku lintas perpustakaan di Makassar. Subjek penelitian meliputi lima pustakawan dari tiga perpustakaan di Makassar dan sepuluh pengguna umum (pelajar, mahasiswa, dan masyarakat) yang dipilih secara purposive untuk menguji kemudahan dan fungsi aplikasi. Stakeholder utama terdiri dari pengembang, pustakawan, dan calon pengguna akhir.

Alat dan Teknik Analisis

Alat bantu yang digunakan dalam proses pengembangan aplikasi i-Read meliputi:

1. Figma sebagai tools perancangan antarmuka pengguna (UI) dan pengalaman pengguna (UX).
2. Android Studio untuk pengembangan aplikasi mobile.
3. Integrasi API dan layanan Firebase sebagai backend aplikasi.
4. Google Maps untuk fitur pencarian lokasi perpustakaan.

Teknik analisis dan pengujian yang dilakukan terdiri dari:

1. Pengujian fungsional, meliputi unit test dan integration test, untuk memastikan keberfungsian fitur aplikasi secara teknis.
2. Pengujian antarmuka pengguna (user interface testing) untuk mengevaluasi kemudahan akses dan pengalaman penggunaan aplikasi.
3. Evaluasi kualitatif melalui wawancara dan kuesioner yang diberikan kepada pustakawan dan pengguna umum sebagai subjek uji.
4. Pengujian aplikasi secara langsung oleh pengguna akhir untuk memastikan aplikasi berfungsi sesuai kebutuhan di lapangan.

Alat dan teknik tersebut diterapkan secara terpadu untuk mendukung pengembangan, pengujian, dan penyesuaian aplikasi i-Read sesuai kebutuhan pengguna, sehingga hasil yang diperoleh optimal dan fungsional.

HASIL DAN DISKUSI

Pada tahap pengujian aplikasi i-Read, sejumlah aspek penting teridentifikasi yang menunjukkan efektivitas aplikasi dalam memfasilitasi pencarian buku di perpustakaan. Pengujian melibatkan beberapa pengguna yang berinteraksi langsung dengan aplikasi untuk mencari buku. Pengembangan aplikasi mengikuti metode Agile dengan empat tahap; perencanaan, analisis kebutuhan, desain, serta implementasi meliputi coding dan pengujian. [15]. Hasil dari setiap tahapan pengembangan Agile ini adalah sebagai berikut:

Hasil Rencana Sistem

Tahap awal dalam pengembangan sistem informasi adalah perencanaan, yang mencakup identifikasi sistem lama dan penentuan pembaruan yang diperlukan melalui system request [16]. Tahap perencanaan dimulai dengan observasi dan wawancara dengan klien untuk mengidentifikasi kebutuhan sistem, yang kemudian dituangkan dalam dokumen permintaan. Diperlukan platform survei berbasis insentif, dan studi kelayakan dilakukan untuk menilai kesiapan sumber daya, potensi keberhasilan, serta risiko yang mungkin timbul.

Tabel 1. Studi Kelayakan Sistem

Studi Kelayakan Sistem Pengumpulan Survey
Studi Kelayakan Teknis
Secara teknis, pengembangan aplikasi i-Read dinilai layak, meskipun terdapat beberapa risiko yang perlu diperhatikan:
<ol style="list-style-type: none">a. Risiko Berhubungan dengan Kefamiliaran dengan Aplikasi: Risiko Sedang<ul style="list-style-type: none">o Tim Pengembang: Sudah familiar dengan pengembangan aplikasi mobile, namun membutuhkan pemahaman lebih tentang integrasi pencarian buku dan navigasi peta.

- Tim Penguji (Pegguna): Penguji harus memastikan aplikasi mudah digunakan dan memberikan umpan balik tentang fungsionalitas yang diperlukan bagi pengguna..
- b. Risiko Berhubungan dengan Kefamiliaran dengan Teknologi: Risiko Rendah
 - Tim Pengembang: Familiar dengan teknologi yang digunakan, seperti Flutter untuk aplikasi Android dan Firebase Firestore untuk pengelolaan data.
 - Tim Penguji (Pegguna): Penguji hanya perlu memahami cara penggunaan aplikasi dan memberikan umpan balik tentang antarmuka dan fungsionalitas.
- c. Risiko terkait ukuran proyek: Dengan tim pengembang 3 orang, proyek diperkirakan selesai dalam 3-4 bulan, mengingat kompleksitas yang dikelola dengan baik.
- d. Kompatibilitas dengan sistem eksternal: Risiko rendah, karena integrasi dengan Google Maps API dan Firebase Firestore berjalan lancar tanpa masalah kompatibilitas signifikan.

Studi Kelayakan Ekonomi

Secara ekonomi, biaya pengembangan aplikasi i-Read diperkirakan moderat karena penggunaan layanan gratis Firebase di tahap awal. ROI diperkirakan tercapai dalam 1 hingga 1,5 tahun, terutama jika diadopsi oleh perpustakaan lokal di Makassar. Potensi monetisasi dapat diperoleh melalui promosi buku, kerja sama dengan lembaga pendidikan, atau program sponsorship literasi.

Studi Kelayakan Organisasi

Secara organisasi, aplikasi i-Read memiliki risiko rendah, karena bertujuan mempermudah akses informasi perpustakaan dan mendukung literasi digital, sesuai dengan inisiatif pendidikan di Makassar. Dukungan dari institusi pendidikan memperkuat kelayakan aplikasi untuk berkelanjutan, dengan harapan beberapa perpustakaan tertarik untuk menjalin kolaborasi jangka panjang.

Berdasarkan hasil studi kelayakan ini, pengembangan aplikasi i-Read dinilai feasible untuk dilanjutkan ke tahap analisis kebutuhan sistem.

Hasil Analisis

Kebutuhan fungsional dan non-fungsional menjadi fokus utama pada tahap analisis dalam pengembangan aplikasi, di mana keduanya dikaji untuk memastikan sistem berjalan sesuai tujuan [17]. Kebutuhan fungsional mencakup fitur pencarian buku dan informasi perpustakaan, sementara kebutuhan non-fungsional meliputi keamanan data, performa aplikasi, dan kemudahan penggunaan, yang dijelaskan lebih rinci di tabel berikut.

Tabel 2. Functional Requirement

Functional Requirement			
No	ID	Fitur	Penjelasan

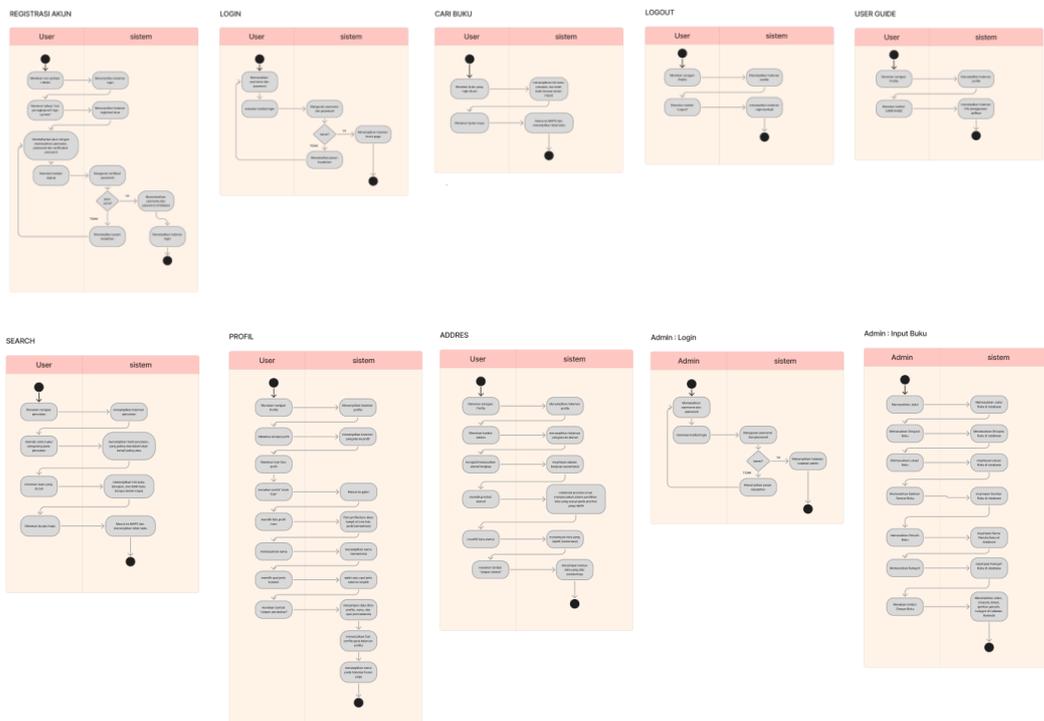
1	FR01	SignUp	Aplikasi ini menyediakan fitur pendaftaran, memungkinkan pengguna membuat akun dengan memasukkan username dan password. Setelah registrasi berhasil, pengguna dapat login untuk mengakses layanan aplikasi.
2	FR02	Login	Aplikasi ini menyediakan fitur pendaftaran, memungkinkan pengguna membuat akun dengan memasukkan username dan password. Setelah registrasi berhasil, pengguna dapat login untuk mengakses layanan aplikasi.
3	FR03	SplashScreen	Ketika aplikasi dibuka, akan ada tampilan awal (splash screen) yang muncul sejenak sebelum pengguna diarahkan ke halaman utama.
4	FR04	Search	Pengguna dapat mencari buku dengan memasukkan kata kunci tertentu, seperti judul buku, nama pengarang, atau kategori buku.
5	FR05	Genre	Pengguna dapat menyaring pencarian buku berdasarkan genre tertentu, seperti fiksi, non-fiksi, sejarah, atau lainnya.
6	FR06	Sinopsis	Setiap buku yang dicari atau direkomendasikan akan memiliki deskripsi singkat yang memberikan gambaran tentang isi buku.
7	FR07	Location	Aplikasi akan menggunakan GPS untuk menunjukkan lokasi perpustakaan atau tempat penyedia buku yang terdekat dengan posisi pengguna.
8	FR08	Profil	Pengguna dapat mengakses profil mereka dan memperbarui bagian alamat sesuai dengan lokasi mereka.
9	FR09	Profile Setting	Aplikasi memungkinkan pengguna untuk mengubah informasi pribadi mereka, termasuk nama, foto profil, dan gender.
10	FR10	Logout	Pengguna bisa keluar dari akun mereka dan login kembali dengan akun yang sama atau akun lain.

Tabel 3. Non Functional Requirement

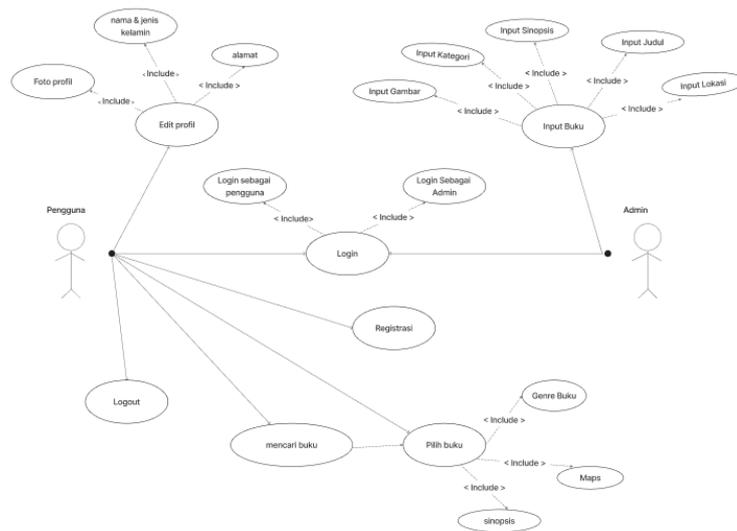
Non Functional Requirement			
No	ID	Fitur	Penjelasan
1	NFR01	Availability	Aplikasi harus tersedia setiap hari dengan downtime maksimal 1 jam per minggu untuk pemeliharaan.
2	NFR02	Reliability	Aplikasi harus dapat menangani minimal 50 pencarian lokasi buku per jam tanpa kegagalan sistem lebih dari 5%
3	NFR03	Ergonomy	Tampilan antarmuka harus sederhana, dan memiliki panduan penggunaan dalam aplikasi agar user dapat memahami cara penggunaan.
4	NFR04	Portability	Aplikasi dikembangkan agar dapat beroperasi pada perangkat Android dengan versi paling rendah 7.0 (Nougat) atau yang lebih baru.
5	NFR05	Memory	Aplikasi ini dapat dijalankan di perangkat dengan RAM minimal 2GB, dan penggunaan memori maksimal 200MB.
6	NFR06	Response time	Pencarian lokasi buku harus menampilkan hasil dalam waktu maksimal 5 detik.
7	NFR07	Safety	N/A

8	NFR08	Security	Akses kedalam aplikasi harus menggunakan autentikasi username dan password.
9	NFR09	Others 1: Bahasa komunikasi	Aplikasi menggunakan 2 bahasa dalam antarmuka, Elemen UI menggunakan bahasa Inggris, sedangkan isi buku dan informasi terkait tetap menggunakan bahasa Indonesia agar mudah di pahami oleh user.
10	NFR10	Others 2 : Branding	Aplikasi harus menampilkan nama dan logo saat aplikasi pertama kali dijalankan selama 2 detik.
11	NFR11	Availability	Aplikasi harus tersedia setiap hari dengan downtime maksimal 1 jam per minggu untuk pemeliharaan.
12	NFR12	Reliability	Aplikasi harus dapat menangani minimal 50 pencarian lokasi buku per jam tanpa kegagalan sistem lebih dari 5%
13	NFR13	Ergonomy	Tampilan antarmuka harus sederhana, dan memiliki panduan penggunaan dalam aplikasi agar user dapat memahami cara penggunaan.
14	NFR14	Portability	Aplikasi harus dapat berjalan di perangkat Android minimal versi 7.0 (Nougat)

Selain mendefinisikan kebutuhan fungsional dan non-fungsional, proses analisis juga sering digambarkan menggunakan diagram UML (Unified Modeling Language), digunakan untuk memodelkan, menjelaskan, dan mengembangkan perangkat lunak [18]. Pada tahap analisis ini, terdapat Use Case Diagram yang mendeskripsikan hubungan antara aktor dan sistem sebagai representasi dari sistem yang dikembangkan [19], serta Activity Diagram yang menunjukkan alur kerja dalam sistem, memastikan bahwa kedua jenis kebutuhan tersebut dapat terintegrasi dengan baik untuk mencapai tujuan sistem yang secara detailnya dapat dilihat pada pada gambar berikut.



Gambar 2. Activity Diagram



Gambar 3. Use Case

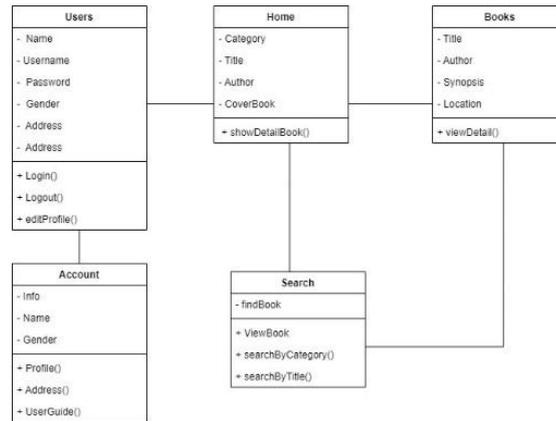
- Use Case Registrasi : use case ini memungkinkan user untuk membuat sebuah akun baru yang dapat digunakan dengan mengisi data diri seperti username dan password
- Use Case Login : use case ini memungkinkan user untuk dapat masuk dan mengakses fitur-fitur yang ada dalam aplikasi dengan menggunakan username dan password
- Use Case Pilih buku : use case ini memungkinkan user untuk memilih buku yang dicari yang didalamnya berisi informasi genre buku, sinopsis, dan maps (lokasi buku)
- Use Case Edit Profil : use case ini memungkinkan user untuk menambahkan/mengatur data diri seperti nama & jenis kelamin, Alamat dan foto profil
- Use Case Mencari buku: use case ini memungkinkan user untuk mencari buku yang ingin dicari di fitur search
- Use Case Logout : use case ini memungkinkan user untuk keluar dari akun dan dapat kembali login dengan akun yang sama ataupun akun yang berbeda
- Use Case Admin - Login : use case ini memungkinkan admin untuk dapat masuk dan menginput data data buku.
- Use Case Admin – Input Buku : Use case ini memungkinkan admin untuk menginput buku, mulai dari judul, synopsis, Lokasi, gambar, penulis dan kategori.

Desain

Hasil dari tahap desain pada penelitian ini berupa rancangan Class Diagram, Sequence Diagram, dan antarmuka pengguna (UI) yang telah disesuaikan dengan kebutuhan serta fitur-fitur yang terdapat dalam aplikasi sistem informasi lokasi perpustakaan, yaitu i-Read.

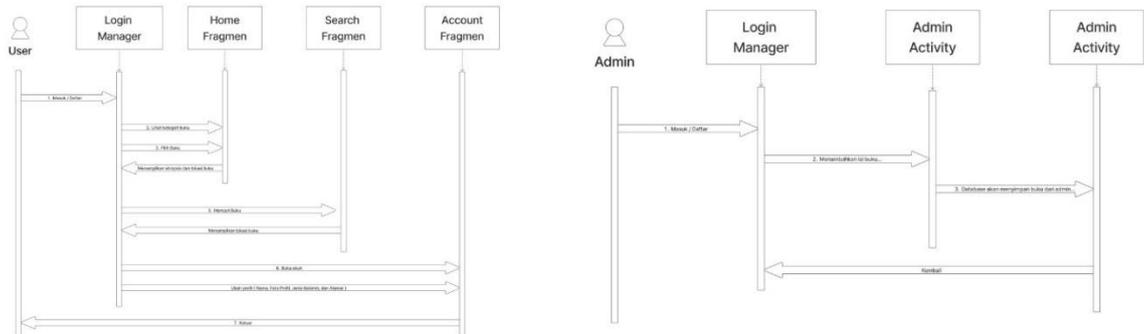
Desain Use Case dan Class Diagram

Use case merupakan representasi fungsi-fungsi dalam suatu sistem yang digambarkan dari perspektif pengguna [20] dibuatlah model menggunakan diagram UML. Class Diagram digunakan untuk menggambarkan struktur data dan hubungan antar kelas yang membentuk aplikasi i-Read.



Gambar 4. Class Diagram

Sementara itu, Sequence diagram menggambarkan interaksi dan pertukaran pesan antar objek dalam suatu sistem [21].



Gambar 5. Sequence Diagram

Kedua diagram ini berfungsi untuk memperjelas desain sistem sebelum tahap implementasi dilakukan.

User Interface (UI)

User Interface (UI) adalah media interaksi antara sistem dan pengguna melalui berbagai perintah seperti penggunaan konten dan input data [22]. Pada aplikasi i-Read, UI dirancang agar interaksi tersebut berlangsung mudah dan efisien, dengan tampilan yang mendukung kenyamanan visual, pencarian buku, akses perpustakaan, serta navigasi lokasi secara intuitif. Seluruh desain UI dikembangkan secara bertahap melalui proses sprint, agar setiap elemen antarmuka dapat dievaluasi dan disesuaikan sesuai kebutuhan pengguna.

1) Sprint pertama

Pada sprint pertama, dikembangkan fitur splash screen, halaman login, dan registrasi. Splash screen tampil sebelum pengguna menuju halaman utama. Fitur login memungkinkan pengguna masuk dengan username dan password, sementara registrasi memungkinkan pembuatan akun baru dengan form username, password, dan verifikasi password. Fitur lupa password belum ada karena belum menjadi prioritas. Setelah login, pengguna diarahkan ke halaman beranda. Berikut Gambar dari splashscreen, login dan signup:



Gambar 6. Sprint 1 (Splash screen, Log In, dan Sign Up)

2) Sprint kedua

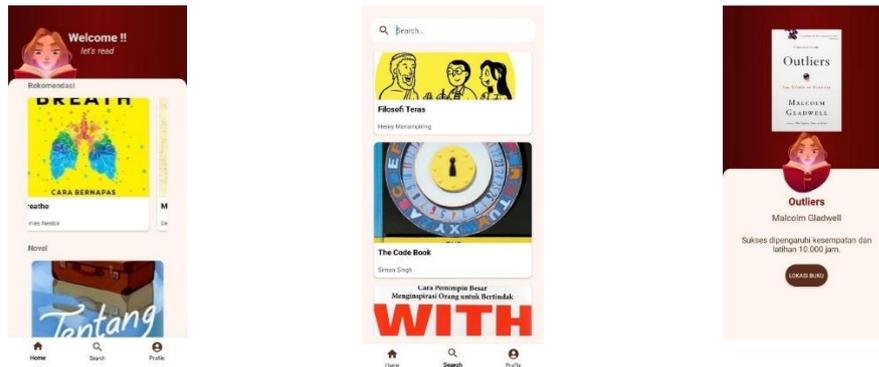
Pada sprint kedua, dikembangkan fitur halaman admin untuk menambah data buku. Fitur ini memungkinkan admin memasukkan informasi lengkap buku, seperti judul, sinopsis, lokasi, penulis, gambar sampul, dan kategori melalui dropdown. Formulir dirancang sederhana dan responsif, dan setelah data terisi, admin dapat menyimpannya dengan tombol "Simpan Buku". Fitur ini penting untuk memperbarui dan menambah koleksi buku di aplikasi. Gambar berikut adalah halaman Admin:



Gambar 7. Sprint 2 (Halaman Admin)

3) Sprint ketiga

Pada sprint ketiga, pengembangan difokuskan pada tiga fitur utama: beranda, pencarian, dan halaman detail buku. Di beranda, pengguna melihat tampilan awal dengan informasi layanan, navigasi utama, dan rekomendasi buku. Fitur pencarian memudahkan pengguna mencari buku berdasarkan judul, pengarang, atau kategori. Setelah menemukan buku, pengguna dapat membuka halaman detail yang menampilkan sinopsis, penerbit, dan lokasi perpustakaan. Fitur-fitur ini dirancang untuk memudahkan pengguna menemukan informasi secara efisien, dengan tampilan yang konsisten dan responsif agar navigasi aplikasi menjadi lebih mudah dan intuitif. Berikut Gambar dari halaman beranda, pencarian dan buku:



Gambar 8. Sprint 3 (Beranda, Pencarian, dan Buku)

4) Sprint keempat

Pada sprint keempat, pengembangan fokus pada fitur pengaturan akun dan panduan penggunaan. Di pengaturan akun, pengguna dapat mengelola informasi pribadi, seperti mengganti foto profil, nama, jenis kelamin, dan alamat tempat tinggal yang dapat digunakan untuk menemukan perpustakaan terdekat. Fitur Guide menyediakan panduan singkat dan jelas tentang cara menggunakan aplikasi, termasuk mencari buku, melihat detail buku, dan navigasi ke lokasi perpustakaan. Fitur-fitur ini bertujuan meningkatkan kenyamanan dan pemahaman pengguna dalam menggunakan aplikasi i-Read. Berikut gambar dari halaman setting profil, alamat dan guide:



Gambar 9. Sprint 4 (Set Profil, Alamat, dan Guide)

Hasil Implementasi

Berikut merupakan pembahasan dari beberapa bagian fungsional dari Aplikasi Iread yang akan diuji dengan menggunakan white-box testing dan Black-box Testing. Pengujian sistem bertujuan mengevaluasi kinerja dan tampilan aplikasi untuk memastikan fungsinya berjalan dengan baik [23].

Hasil Pengujian *White Box*

Pengujian white-box merupakan metode yang digunakan untuk memeriksa dan menganalisis kode program secara langsung guna mendeteksi adanya kesalahan dalam logika

atau struktur program [24]. Pendapat lain menyatakan bahwa metode ini berfokus sepenuhnya pada pemeriksaan kode sumber tanpa memperhatikan tampilan antarmuka pengguna dari aplikasi yang diuji [25]. Berikut merupakan pembahasan White-box untuk fitur login dan register.

1. Halaman Login

```

public class Login extends AppCompatActivity {

    2 usages
    private EditText etUsername, etPassword;
    2 usages
    private Button btnLogin;
    4 usages
    private ImageView eyePassword;
    private TextView signUpHere;
    2 usages
    private FirebaseFirestore db;
    1 usage
    private Handler handler = new Handler(); // Handler untuk delay

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login); // Pastikan layout ini ada

        db = FirebaseFirestore.getInstance();

        eyePassword = findViewById(R.id.eyePassword);
        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        btnLogin = findViewById(R.id.btnLogin);
        signUpHere = findViewById(R.id.signUpText);

        eyePassword.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                etPassword.setInputType(InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD);
                eyePassword.setImageResource(R.drawable.baseline_remove_red_eye_24);

                eyePassword.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        etPassword.setInputType(InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD);
                        eyePassword.setImageResource(R.drawable.baseline_remove_red_eye_24);

                        handler.postDelayed(new Runnable() {
                            @Override
                            public void run() {
                                etPassword.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_TEXT_VARIATION_PASSWORD);
                                eyePassword.setImageResource(R.drawable.baseline_remove_red_eye_24);
                                etPassword.setSelection(etPassword.getText().length());
                            }
                        }, delayMillis: 600); // 600ms
                    }
                });
            }
        });

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String username = etUsername.getText().toString().trim();
                String password = etPassword.getText().toString().trim();

                if (username.isEmpty() || password.isEmpty()) {
                    Toast.makeText(getApplicationContext(), text: "Username atau Password kosong!", Toast.LENGTH_SHORT).show();
                } else if (username.equals("admin") && password.equals("admin123")) {
                    // Jika login sebagai admin
                    Toast.makeText(context: Login.this, text: "Selamat datang Admin", Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent( packageContext: Login.this, Admin.class);
                    startActivity(intent);
                    finish();
                } else {
                    // Kalau bukan admin, cek di database
                    checkAkun(username, password);
                }
            }
        });

        signUpHere.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent( packageContext: Login.this, Signup.class);
                startActivity(intent);
            }
        });
    }
}

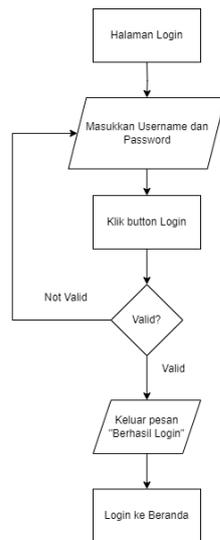
```

```
private void checkAkun(String username, String password) {
    db.collection( collectionPath: "users") CollectionReference
    .whereEqualTo( field: "username", username) Query
    .get() Task<QuerySnapshots>
    .addOnSuccessListener(queryDocumentSnapshots -> {
        if (!queryDocumentSnapshots.isEmpty()) {
            // Jika akun ditemukan, kita ambil password yang tersimpan
            String storedPassword = queryDocumentSnapshots.getDocuments().get(0).getString( field: "password");
            if (storedPassword != null && storedPassword.equals(password)) {
                Toast.makeText( context: Login.this, text: "Berhasil Login", Toast.LENGTH_SHORT).show();

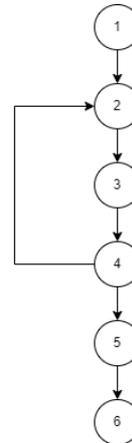
                // Tambahkan: Simpan user_id ke SharedPreferences
                String userId = queryDocumentSnapshots.getDocuments().get(0).getId();
                SharedPreferences prefs = getSharedPreferences( name: "user_session", Context.MODE_PRIVATE);
                SharedPreferences.Editor editor = prefs.edit();
                editor.putString("user_id", userId);
                editor.apply();

                Intent intent = new Intent( packageContext: Login.this, Home.class);
                startActivity(intent);
                finish();
            } else {
                Toast.makeText( context: Login.this, text: "Password salah", Toast.LENGTH_SHORT).show();
            }
        } else {
            // Akun tidak ditemukan, log info untuk debugging
            Toast.makeText( context: Login.this, text: "Akun tidak ditemukan", Toast.LENGTH_SHORT).show();
            Log.d( tag: "Login", msg: "Akun tidak ditemukan untuk username: " + username);
        }
    })
    .addOnFailureListener(e -> {
        // Log error jika terjadi kegagalan query
        Log.e( tag: "Login", msg: "Gagal mencari akun di Firestore", e);
        Toast.makeText( context: Login.this, text: "Gagal login, coba lagi", Toast.LENGTH_SHORT).show();
    });
}
```

Gambar 10. Coding Login



Gambar 11. Flowchart Login



Gambar 12. Flowgraph Login

$$V(G) = (E - N) + 2$$

V(G) = Jumlah Region

E = Jumlah edge yang ditentukan dengan gambar panah N = Jumlah simpul grafik (node) dengan gambar lingkaran

$$V(G) = (6 - 6) + 2$$

$$V(G) = 2$$

Jalur 1 = 1-2-3-4-5-6

Jalur 2 = 1-2-3-4-2-3-4-5-6

Tabel 4. Pengujian Path Fungsi Login

Path	1
Jalur	1-2-3-4-5-6
Skenario	1. Halaman Login 2. Masukkan username dan password 3. Klik tombol Login 4. Validasi input (valid) 5. Tampilkan pesan "Berhasil Login" 6. Masuk ke halaman Beranda
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	1. Halaman Login 2. Masukkan username dan password yang salah 3. Klik tombol Login 4. Validasi input (not valid) 2. Masukkan ulang username dan password yang benar 3. Klik tombol Login 4. Validasi input (valid) 5. Tampilkan pesan "Berhasil Login" 6. Masuk ke halaman Beranda
Hasil Pengujian	Berhasil

Setelah jumlah jalur independen ditentukan, langkah selanjutnya adalah membandingkannya dengan tingkat risiko menggunakan tabel perbandingan antara kompleksitas siklomatik dan level risiko[24]. seperti ditunjukkan pada tabel berikut:

Tabel 5. Resiko dari tingkat Prosedur Aplikasi

Nilai	Tipe Prosedur	Tingkat Risiko
1-4	Prosedur Sederhana	Rendah
5-10	Prosedur yang terstruktur dengan baik dan stabil	Rendah
11-20	Prosedur yang lebih kompleks	Menengah
21-50	Prosedur yang kompleks dan kritis	Menengah
>50	Rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji	Sangat Tinggi

Berdasarkan tabel diatas, fungsi Login memiliki tingkat risiko rendah dan prosedur sederhana karena hanya memiliki 2 jalur independen. Setelah itu, langkah berikutnya adalah pembuatan test case [24], Perhatikan Tabek 4.3 berikut :

Tabel 6. Test Case Login

No	Nama Skenario	Kegiatan	Hasil yang di harapkan	Hasil	Keterangan
1.	Uji Login 1	User memasukkan username dan password yang valid	Berhasil login dan dialihkan ke halaman	Sistem menampilkan pesan sukses	

		lalu menekan tombol login	utama, serta menampilkan pesan "Berhasil Login"	dan mengarahkan user ke halaman beranda Sistem	Valid
2.	Uji Login 2	User memasukkan username/password salah lalu mencoba kembali dengan data yang valid	Awalnya muncul pesan error, setelah data valid dimasukkan sistem menampilkan pesan "Berhasil Login"	Sistem menampilkan error terlebih dahulu, lalu berhasil login saat data benar	Valid

2. Halaman Register

```

public class Signup extends AppCompatActivity {

    2 usages
    private EditText etUsername, etPassword, etPasswordVerification, etName, etAddress;
    2 usages
    private Spinner spinnerGender;
    2 usages
    private Button btnSignUp;
    3 usages
    private ImageView eyePassword, eyePasswordVerification, imgProfile;
    no usages
    private Handler handler = new Handler();
    2 usages
    private FirebaseFirestore db;
    3 usages
    private String encodedImage = ""; // Menyimpan gambar dalam format base64

    2 usages
    private static final int PICK_IMAGE_REQUEST = 1; // Kode untuk request galeri

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_signup);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
        // inisialisasi editText, spinner, button, dan imageView
        etUsername = findViewById(R.id.etUsername);
        etPassword = findViewById(R.id.etPassword);
        etPasswordVerification = findViewById(R.id.etPasswordVerification);
        etName = findViewById(R.id.etName);
        etAddress = findViewById(R.id.etAddress);
        spinnerGender = findViewById(R.id.spinnerGender);
        btnSignUp = findViewById(R.id.btnSignUp);
        eyePassword = findViewById(R.id.eyePassword);
        eyePasswordVerification = findViewById(R.id.eyePasswordVerification);
        imgProfile = findViewById(R.id.imageViewProfile);

        // Menangani klik untuk memilih foto profil
        imgProfile.setOnClickListener(v -> openGallery());

        // Mengatur visibilitas password
        eyePassword.setOnClickListener(v -> togglePasswordVisibility(etPassword, eyePassword));
        eyePasswordVerification.setOnClickListener(v -> togglePasswordVisibility(etPasswordVerification, eyePasswordVerification));

        btnSignUp.setOnClickListener(v -> {
            String username = etUsername.getText().toString().trim();
            String password = etPassword.getText().toString().trim();
            String verificationPassword = etPasswordVerification.getText().toString().trim();
            String name = etName.getText().toString().trim();
            String address = etAddress.getText().toString().trim();
            String gender = spinnerGender.getSelectedItem().toString();

            if (username.isEmpty() || password.isEmpty() || verificationPassword.isEmpty() || name.isEmpty() || address.isEmpty() || encodedImage.isEmpty())
                Toast.makeText(context, Signup.this, "Harap isi semua kolom!", Toast.LENGTH_SHORT).show();
            else {
                if (password.length() < 8) {
                    Toast.makeText(context, Signup.this, "Password harus minimal 8 karakter!", Toast.LENGTH_SHORT).show();
                } else if (!password.equals(verificationPassword)) {
                    Toast.makeText(context, Signup.this, "Password tidak sama. Silakan coba lagi.", Toast.LENGTH_SHORT).show();
                } else {
                    buatAkun(username, password, name, address, gender, encodedImage);
                }
            }
        });
    }
}

```

```

private void openGallery() {
    Intent intent = new Intent(Intent.ACTION_PICK);
    intent.setType("image/*");
    startActivityForResult(intent, PICK_IMAGE_REQUEST);
}

// Menangani hasil pemilihan gambar dari galeri
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data != null) {
        try {
            // Mendapatkan gambar yang dipilih dari galeri
            InputStream inputStream = getContentResolver().openInputStream(data.getData());
            Bitmap bitmap = BitmapFactory.decodeStream(inputStream);

            // Menampilkan gambar di ImageView
            imgProfile.setImageBitmap(bitmap);

            // Mengonversi gambar menjadi format base64
            encodedImage = encodeImageToBase64(bitmap);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}

private String encodeImageToBase64(Bitmap bitmap) {
    ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, 100, byteArrayOutputStream);
    byte[] byteArray = byteArrayOutputStream.toByteArray();
    return Base64.encodeToString(byteArray, Base64.DEFAULT);
}

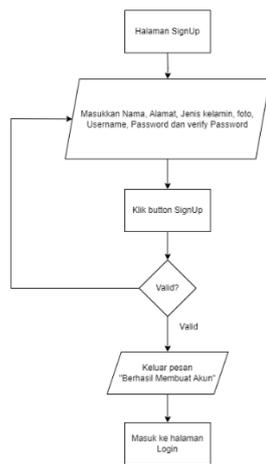
// Fungsi untuk mengganti visibilitas password
2 usages
private void togglePasswordVisibility(EditText etPassword, ImageView eyeIcon) {
    if (etPassword.getInputType() == InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD) {
        etPassword.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_TEXT_VARIATION_PASSWORD);
        eyeIcon.setImageResource(R.drawable.baseline_remove_red_eye_24);
    } else {
        etPassword.setInputType(InputType.TYPE_TEXT_VARIATION_VISIBLE_PASSWORD);
        eyeIcon.setImageResource(R.drawable.baseline_remove_red_eye_24);
    }
    etPassword.setSelection(etPassword.getText().length());
}

private void buatAkun(String username, String password, String name, String address, String gender, String profilePhoto) {
    Map<String, Object> user = new HashMap<>();
    user.put("username", username);
    user.put("password", password);
    user.put("name", name);
    user.put("address", address);
    user.put("gender", gender);
    user.put("profilePhoto", profilePhoto);

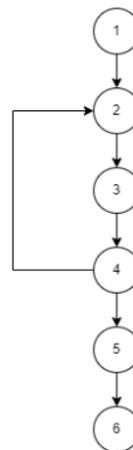
    db.collection("users").collectionReference
        .add(user) Task<DocumentReference> -> {
        .addOnSuccessListener(documentReference -> {
            Toast.makeText(context: Signup.this, text: "Berhasil membuat akun", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(context: Signup.this, Login.class);
            startActivity(intent);
            finish();
        })
        .addOnFailureListener(e -> Toast.makeText(context: Signup.this, text: "Gagal membuat akun", Toast.LENGTH_SHORT).show());
    }
}

```

Gambar 13. Code Register



Gambar 14. Flowchart Register



Gambar 15. Flowchart register

$$V(G) = (E - N) + 2$$

V(G) = Jumlah Region

E= Jumlah edge yang ditentukan dengan gambar panah N= Jumlah simpul grafik (node) dengan gambar lingkaran

$$V(G) = (6-6) + 2$$

$$V(G) = 2$$

Jalur 1= 1-2-3-4-5-6

Jalur 2= 1-2-3-4-2-3-4-5-6

Tabel 7. Pengujian Path Fungsi Register

Path	1
Jalur	1-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Halaman SignUp 2. Masukkan semua data dengan lengkap (nama, alamat, jenis kelamin, foto, username, password, Verify Password.) 3. Klik tombol SignUp 4. Validasi input (valid) 5. Muncul pesan "Berhasil Membuat Akun" 6. Dialihkan ke halaman login
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<ol style="list-style-type: none"> 1. Halaman SignUp 2. Masukkan data yang tidak lengkap 3. Klik tombol SignUp 4. Validasi input (tidak valid) 2. Masukkan ulang semua data dengan benar 3. Klik tombol SignUp 4. Validasi input (valid) 5. Muncul pesan "Berhasil Membuat Akun" 6. Dialihkan ke halaman login
Hasil Pengujian	Berhasil

Setelah diketahui jumlah jalur independennya, maka akan dilakukan perbandingan menggunakan tabel hubungan antara cyclomatic complexity dan resiko pada tabel berikut:

Tabel 8. Resiko dari tingkat Prosedur Aplikasi

Nilai	Tipe Prosedur	Tingkat Resiko
1-4	Prosedur Sederhana	Rendah
5-10	Prosedur yang terstruktur dengan baik dan stabil	Rendah
11-20	Prosedur yang lebih kompleks	Menengah
21-50	Prosedur yang kompleks dan kritis	Menengah
>50	Rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji	SangatTinggi

Berdasarkan tabel sebelumnya, fungsi Register dikategorikan memiliki risiko rendah serta prosedur yang sederhana, karena hanya terdapat dua jalur independen. Setelah jumlah jalur

independen diketahui, tahap berikutnya adalah penyusunan test case, yang ditampilkan dalam tabel di bawah ini.

Tabel 9. Test Case Register

No	Nama Skenario	Kegiatan	Hasil yang di harapkan	Hasil	Keterangan
1.	Uji Register 1	User mengisi semua kolom seperti nama, alamat, jenis kelamin, foto, username, password, verify password, lalu klik tombol SignUp	Berhasil register dan menampilkan pesan "Berhasil Membuat Akun" serta dialihkan ke halaman login	Sistem menampilkan pesan sukses dan mengarahkan user ke halaman login	Valid
2.	Uji Register 2	User mengisi data tidak lengkap lalu klik SignUp 'muncul pesan error' kemudian mengisi ulang seluruh kolom dengan benar lalu klik SignUp	Awalnya sistem menampilkan error input kosong. Setelah diisi ulang, sistem menampilkan pesan sukses dan redirect ke halaman login	Sistem menampilkan pesan error terlebih dahulu, kemudian menampilkan pesan sukses	Valid

Hasil Pengujian *Black Box*

Pengujian aplikasi i-Read dilakukan menggunakan metode Black Box, di mana pengujian difokuskan pada fungsionalitas aplikasi tanpa melihat struktur kode program [23]. Pengujian ini melibatkan pengguna akhir untuk mengevaluasi apakah fitur-fitur seperti login, pencarian buku, informasi lokasi perpustakaan, serta pengelolaan profil pengguna berjalan sesuai harapan. Tujuan dari pengujian ini adalah untuk mengidentifikasi kesalahan dari sudut pandang pengguna. Rangkaian kasus uji tersebut dapat dilihat pada Tabel 1.3 berikut.

Tabel 10. Pengujian Sistem

Pengujian Black Box				
Kode Uji	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Diperoleh	Status
UB-1	Login dengan username dan password kosong	Pesan error "Username atau password kosong" ditampilkan	Pesan error "Username atau password kosong" tampil	Berhasil
UB-2	Login dengan username salah	Pesan error "Akun tidak ditemukan" ditampilkan	Pesan error "Akun tidak ditemukan" tampil	Berhasil
UB-3	Login dengan Password salah	Pesan error "Password salah" ditampilkan	Pesan error "Password salah" ditampilkan	Berhasil
UB-4	Login dengan username dan password benar	Pesan "Login berhasil" muncul serta pengguna berhasil masuk dan diarahkan ke home	Pesan "Login berhasil" muncul serta pengguna berhasil masuk dan diarahkan ke home	Berhasil

UB-5	Pendaftaran dengan semua data lengkap	Pesan "Pendaftaran berhasil" dan diarahkan ke halaman login	Pesan "Pendaftaran berhasil" tampil dan diarahkan ke login	Berhasil
UB-6	Pendaftaran tanpa mengisi salah satu data wajib	Pesan error "Semua data wajib diisi" ditampilkan	Pesan error "Semua data wajib diisi" ditampilkan	Berhasil
UB-7	Mencari buku berdasarkan judul yang tersedia	Buku muncul dalam daftar pencarian	Buku ditemukan dan tampil dalam hasil pencarian	Berhasil
UB-8	Mencari buku yang tidak tersedia di database	Pesan "Buku tidak ditemukan" ditampilkan	Pesan "Buku tidak ditemukan" tampil	Berhasil
UB-9	Menampilkan rekomendasi buku secara acak	Daftar buku tampil secara acak dan berubah saat pengguna membuka kembali fitur	Daftar rekomendasi tampil acak dan berubah saat dimuat ulang	Berhasil
UB-10	Menampilkan detail buku (judul, penulis, sinopsis, kategori, dan lokasi)	Informasi judul, penulis, sinopsis, kategori, dan lokasi buku ditampilkan lengkap	Informasi judul, penulis, sinopsis, kategori, dan lokasi tampil	Berhasil
UB-11	Menampilkan informasi lokasi buku	Lokasi perpustakaan ditampilkan dan dapat diklik untuk diarahkan ke perpustakaan	Lokasi buku tampil dan dapat diklik untuk membuka arah perpustakaan	Berhasil
UB-12	Mengubah profil (nama pengguna, foto, dan gender)	Pesan "Profil berhasil diperbarui" ditampilkan	Pesan "Profil berhasil diperbarui" tampil	Berhasil
UB-13	Mengubah alamat pengguna	Pesan "Alamat berhasil diperbarui" ditampilkan	Pesan "Alamat berhasil diperbarui" tampil	Berhasil
UB-14	Logout dari aplikasi	Pengguna diarahkan ke halaman login	Pengguna berhasil logout dan diarahkan ke halaman logi	Berhasil

Diskusi

Dari perbandingan fitur dan efektivitas sistem informasi perpustakaan yang telah diteliti, aplikasi i-Read memiliki keunggulan utama dalam pencarian lokasi buku lintas perpustakaan dengan sistem peta terintegrasi. Berbeda dengan penelitian sebelumnya yang lebih berfokus pada manajemen koleksi buku dalam satu institusi, i-Read memberikan solusi pencarian yang lebih luas dengan menghubungkan berbagai perpustakaan di wilayah Makassar.

Dari segi fitur, sistem perpustakaan seperti SDN Cawang 12 Pagi [6], SDK Buntal [3], SMK Negeri 1 Larantuka [8], dan SMA Islam Al-Mizan [5] umumnya berfokus pada manajemen data buku dan anggota, pencatatan transaksi peminjaman dan pengembalian, serta laporan perpustakaan. Penelitian [10] ini membahas berbagai jenis sistem informasi yang dapat

diterapkan dalam perpustakaan, seperti Transaction Processing System (TPS), Management Information System (MIS), Decision Support System (DSS), Office Automation System (OAS), dan Intelligent Support System (ISS). Sistem perpustakaan SMA Negeri 6 Depok [9] memiliki fitur utama seperti manajemen data buku dan anggota, otomatisasi peminjaman dan pengembalian, pelaporan berkala, serta keamanan dengan autentikasi pengguna, yang meningkatkan efisiensi dan akurasi dalam pengelolaan perpustakaan. Sementara itu, i-Read menawarkan fitur unik seperti pencarian berdasarkan lokasi dan integrasi dengan Google Maps, memungkinkan pengguna menemukan buku yang mereka butuhkan di berbagai perpustakaan dalam kota tanpa harus mendatangi tiap tempat secara langsung.

Efektivitas sistem juga menunjukkan perbedaan yang signifikan. Penelitian sebelumnya lebih menekankan pada pengelolaan internal perpustakaan, seperti otomasi laporan di SMA Negeri 6 Depok [9], optimasi pencarian buku dengan algoritma Levenshtein Distance di SMA Islam Al-Mizan [5], serta penggunaan barcode untuk transaksi di SMK Negeri 1 Larantuka [8]. Sistem SDN Cawang 12 Pagi [6] meningkatkan efisiensi perpustakaan dengan otomatisasi pencatatan peminjaman, pengembalian, dan laporan. Penghitungan denda otomatis serta pengujian Blackbox Testing memastikan akurasi data, sehingga pengelolaan perpustakaan lebih sistematis dibanding metode manual. Secara efektivitas, penelitian [10] menunjukkan bahwa perpustakaan dapat lebih optimal jika menggunakan kombinasi sistem informasi yang mencakup pengolahan transaksi, manajemen data, analisis keputusan, serta otomatisasi dokumen dan komunikasi. Sebaliknya, i-Read meningkatkan aksesibilitas pencarian buku dengan cakupan lebih luas, memberikan solusi bagi pengguna yang kesulitan menemukan buku karena keterbatasan stok di satu perpustakaan. Dengan penyediaan informasi ketersediaan buku secara real-time, aplikasi ini tidak hanya menghemat waktu pencarian tetapi juga mendukung peningkatan budaya literasi di lingkungan akademik.

Secara keseluruhan, dibandingkan dengan enam penelitian sebelumnya, i-Read menawarkan pendekatan inovatif yang memperluas cakupan pencarian buku dari sistem yang terbatas pada satu institusi menjadi sistem yang terhubung antar perpustakaan. Fitur pencarian berbasis lokasi menjadi nilai tambah utama yang membedakannya dari sistem lain yang lebih berfokus pada manajemen koleksi dan transaksi peminjaman. Dengan metode pengembangan Agile, aplikasi i-Read juga lebih adaptif terhadap kebutuhan pengguna, memungkinkan pembaruan dan penyempurnaan fitur secara cepat berdasarkan umpan balik pengguna.

KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang dan mengimplementasikan aplikasi i-Read sebagai solusi atas masalah sulitnya pencarian buku di berbagai perpustakaan di Makassar. Dengan menggunakan metode Agile, aplikasi ini mengintegrasikan fitur pencarian lokasi buku, tampilan antarmuka yang sederhana, serta sistem informasi perpustakaan yang adaptif terhadap kebutuhan pengguna. Temuan penelitian menunjukkan bahwa aplikasi i-Read mampu meningkatkan efisiensi pencarian buku dan memperluas akses masyarakat terhadap literatur yang tersebar di berbagai lokasi.

Kontribusi utama dari penelitian ini terletak pada pendekatan baru dalam pengembangan sistem informasi perpustakaan yang bersifat lintas institusi, memungkinkan integrasi koleksi dari berbagai perpustakaan dalam satu platform terpusat. Hal ini menjadi novelty utama dari aplikasi i-Read, karena menghadirkan solusi terpadu yang belum banyak diimplementasikan di tingkat lokal.

Sebagai saran, pengembangan lanjutan dapat mencakup fitur peminjaman buku online, pelacakan ketersediaan stok, serta integrasi dengan katalog perpustakaan nasional. Selain itu, aplikasi ini membuka peluang besar untuk kolaborasi antar institusi pendidikan di Makassar dalam membentuk ekosistem literasi digital yang inklusif dan berkelanjutan. Dengan demikian, i-Read berpotensi menjadi model rujukan dalam pengelolaan perpustakaan digital yang modern dan terintegrasi.

REFERENCES

- [1] A. Hidayat and D. A. R, "DEPOK DENGAN QR CODE," vol. 28, no. 3, pp. 228–242, 2023.
- [2] Sucianti, Nilam Permatasari Munir, and Nur Fakhrunnisaa, "Pengembangan Media Pembelajaran Scrapbook Berbasis Kontekstual pada Materi Bagian dan Fungsi Tumbuhan Kelas IV SDN 21 Tadette," *J. Vocat. Informatics Comput. Educ.*, vol. 1, no. 2, pp. 72–78, 2023, doi: 10.61220/voice.v1i2.20238.
- [3] W. Harjono and Kristianus Jago Tute, "Perancangan Sistem Informasi Perpustakaan Berbasis Web Menggunakan Metode Waterfall," *SATESI J. Sains Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 47–51, 2022, doi: 10.54259/satesi.v2i1.773.
- [4] J. Isa and A. Thomas, "Sistem Informasi Perpustakaan Digital Berbasis Web Pada STMIK Catur Sakti Kendari," *Sist. Inf. dan Tek. Komput.*, vol. 7, no. 2, pp. 177–181, 2022.
- [5] D. Abdurahman and Deffy Susanti, "Perancangan Aplikasi Sistem Informasi Perpustakaan Menggunakan Algoritma Levenshtein Distance Di Perpustakaan Sma Islam Al-Mizan," *INFOTECH J.*, vol. 9, no. 2, pp. 371–376, 2023, doi: 10.31949/infotech.v9i2.6263.
- [6] A. D. Pangestu and L. A. Utami, "Sistem Informasi Perpustakaan Berbasis Web Pada Sdn Cawang 12 Pagi," *IJIS - Indones. J. Inf. Syst.*, vol. 7, no. 1, pp. 25–34, 2022, doi: 10.36549/ijis.v7i1.196.
- [7] D. Penerapan and M. Fifo, "1* , 2 1,2," pp. 17–23, 2023.
- [8] B. S. D. Kedang, K. Septianzah, and N. Farkhatin, "Implementasi Sistem Informasi Perpustakaan pada SMK Negeri 1 Larantuka Berbasis Java," *J. Ris. dan Apl. Mhs. Inform.*, vol. 4, no. 02, pp. 256–263, 2023, doi: 10.30998/jrami.v4i02.8078.
- [9] S. Yulianti, "Perancangan Aplikasi Perpustakaan pada Sekolah Menengah Atas Negeri 6 Depok Berbasis Java," *J. Ris. dan Apl. Mhs. Inform.*, vol. 2, no. 01, pp. 84–90, 2021, doi: 10.30998/jrami.v2i01.798.
- [10] A. O. P. Dewi, "Pengembangan Sistem Informasi Perpustakaan untuk Mengelola Data Perpustakaan," *Anuva J. Kaji. Budaya, Perpustakaan, dan Inf.*, vol. 5, no. 2, pp. 213–220, 2021, doi: 10.14710/anuva.5.2.213-220.
- [11] Hendra Sopandi and Novita Anggraini, "Aplikasi Perpustakaan Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus: Smpn 3 Pacet)," *Nuansa Inform.*, vol. 18, no. 1, pp. 41–52, 2024, doi: 10.25134/ilkom.v18i1.43.
- [12] E. Sitompul and E. Hutabri, "Perancangan Aplikasi Resep Masakan Tradisional Indonesia Menggunakan Pendekatan Agile Process Dengan Model Extreme Programming Berbasis Android," *Comput. Sci. Ind. Eng.*, vol. 9, no. 8, 2023, doi: 10.33884/comasiejournal.v9i8.7976.

-
- [13] R. Noveandini, M. S. Wulandari, and A. Hakim, "Penerapan Metode Scrum Pada Rancang Bangun Sistem Informasi Penjualan Toko Sepatu Rabbani Shoes," *Semin. Nas. Teknol. Inf. dan Komun. STI&K*, vol. 7, no. 1, pp. 192–198, 2023.
- [14] M. Y. Suntana, T. Terttiaavini, P. Oktariani, F. Sanawi, M. D. Akbar, and A. W. Pratama, "Perancangan Aplikasi Solusi Publik Untuk Pengaduan Masyarakat Mengenai Fasilitas Umum," *J. Inf. Syst. Manag. Digit. Bus.*, vol. 1, no. 3, pp. 332–334, 2024, doi: 10.59407/jismdb.v1i3.738.
- [15] N. Hikmah, A. Suradika, and R. A. Ahmad Gunadi, "Metode Agile Untuk Meningkatkan Kreativitas Guru Melalui Berbagi Pengetahuan (Knowledge Sharing) (Studi Kasus: Sdn Cipulir 03 Kebayoran Lama, Jakarta)," *Instruksional*, vol. 3, no. 1, p. 30, 2021, doi: 10.24853/instruksional.3.1.30-39.
- [16] V. A. Z. Sodikin, Reni Amaranti, and Djamaludin, "Perancangan Sistem Informasi Manajemen Gudang PT. X," *J. Ris. Tek. Ind.*, vol. 1, no. 1, pp. 58–67, 2021, doi: 10.29313/jrti.v1i1.141.
- [17] N. Purwandari and A. Fauzi, "Perancangan Sistem Informasi Manajemen Pada Toko XYZ Berbasis Desktop," *J. Sist. Inf. Bisnis*, vol. 1, no. 2, pp. 54–64, 2020, doi: 10.55122/junsibi.v1i2.171.
- [18] C. A. Ayu Binangkit, A. Voutama, and N. Heryana, "Pemanfaatan Uml (Unified Modeling Language) Dalam Perencanaan Sistem Pengelolaan Sewa Alat Musik Berbasis Website," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 2, pp. 1429–1436, 2023, doi: 10.36040/jati.v7i2.6858.
- [19] G. F. Fitriana, "Penguujian Aplikasi Pengenalan Tulisan Tangan menggunakan Model Behaviour Use case," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 2, pp. 200–213, 2020, doi: 10.35957/jatisi.v7i2.390.
- [20] L. Setiyani, "Desain Sistem : Use Case Diagram Pendahuluan," *Pros. Semin. Nas. Inov. Adopsi Teknol. 2021*, no. September, pp. 246–260, 2021, [Online]. Available: <https://journal.uui.ac.id/AUTOMATA/article/view/19517>
- [21] S. Nabila, A. R. Putri, A. Hafizhah, F. H. Rahmah, and R. Muslikhah, "Pemodelan Diagram UML Pada Perancangan Sistem Aplikasi Konsultasi Hewan Peliharaan Berbasis Android (Studi Kasus: Alopét)," *J. Ilmu Komput. dan Bisnis*, vol. 12, no. 2, pp. 130–139, 2021, doi: 10.47927/jikb.v12i2.150.
- [22] M. S. Hartawan, "Penerapan User Centered Design (Ucd) Pada Wireframe Desain User Interface Dan User Experience Aplikasi Sinopsis Film," *Jeis J. Elektro Dan Inform. Swadharma*, vol. 2, no. 1, pp. 43–47, 2022, doi: 10.56486/jeis.vol2no1.161.
- [23] A. Praniffa *et al.*, "Penguujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web Black Box And White Box Testing Of Web-Based Parking Information System," *J. Test. Dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 1–16, 2023.
- [24] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. Abu Bakar, N. Novianti, and R. Ramadan, "Penguujian White-Box Pada Aplikasi Debt Manager Berbasis Android," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 4, no. 1, pp. 11–22, 2022, doi: 10.53580/naratif.v4i1.147.
- [25] A. A. Gusdevi, Harya; Hutama, Brian Damastu Ridho; Rizaldi, "View of PENGUJIAN WHITE BOX PADA APLIKASI PINJAM DULU SERATUS BERBASIS WEBSITE.pdf"