

# Pengembangan Aplikasi Mobile Berbasis Android untuk Manajemen Antrian Bimbingan KP dan Proyek Akhir dengan Memanfaatkan Fitur Location Based Service

Nesya Anfasha Rosa<sup>1\*</sup>, Shumaya Resty Ramadhani<sup>2\*</sup>

\* Teknik Informatika, Politeknik Caltex Riau

[nesya17ti@mahasiswa.pcr.ac.id](mailto:nesya17ti@mahasiswa.pcr.ac.id)<sup>1</sup>, [shumaya@pcr.ac.id](mailto:shumaya@pcr.ac.id)<sup>2</sup>

## Article Info

### Article history:

Received 2021-09-16

Revised 2022-05-17

Accepted 2022-06-23

### Keyword:

*Android,  
Antrian Prioritas,  
Bimbingan Mahasiswa,  
Location Based Service.*

## ABSTRACT

Guidance is a procedure that must be done by students to get assistance and supervision in completing work practice reports and final assignments. This activity is carried out by holding meetings between students and the lecturers concerned. Students need to arrange a schedule by contacting the lecturer via communication media before conducting guidance. After arranging the schedule with the lecturer, students must be on time according to the predetermined schedule. Even though the tutoring schedule has been set, there are still students who come late. This causes guidance time to be wasted and other students do not get a schedule, so students are constrained in completing their reports. Therefore, an Android-based queue management application was developed by utilizing location-based service technology. This application uses priority-based queue management and location-based services that are used to detect lecturer geofence areas. Tests have been carried out on applications built using black-box testing techniques, with 100% results that all functional requirements have been successfully implemented. Likewise with the results of white-box testing that the program code has been executed with a percentage of 100%. In addition, testing using usability testing was also carried out with an average test percentage of 85.4% of respondents agreeing that the queuing application had met the criteria in the statement.



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

## I. PENDAHULUAN

Bimbingan merupakan sarana bagi mahasiswa untuk mendapatkan arahan, bantuan, saran, dan koreksi atas skripsi atau laporan yang sedang disusun dengan bantuan dosen yang bersangkutan [1]. Untuk melakukan bimbingan, mahasiswa perlu mengatur jadwal bersama dosen pada tanggal dan waktu yang ditentukan. Peneliti melakukan survey menggunakan kuesioner kepada mahasiswa dan dosen Jurusan Teknologi Informasi Politeknik Caltex Riau yang pernah melakukan bimbingan. Hasil survey menyimpulkan bahwa kendala yang dihadapi oleh dosen pada saat proses bimbingan yaitu mahasiswa yang tidak datang tepat waktu. Pada kondisi ini, terdapat dosen yang bersedia menunggu kedatangan mahasiswa dengan sedikit toleransi waktu, lalu segera memberikan bimbingan. Namun, bagi sebagian dosen hal ini dapat mengganggu jadwal dosen tersebut. Sedangkan kendala

yang dihadapi oleh mahasiswa ialah ketika dosen tidak memberikan konfirmasi bahwa bimbingan tidak dilakukan pada waktu yang sudah ditentukan ataupun dosen tidak dapat memberikan bimbingan karena satu dan lain hal. Sehingga, mahasiswa yang sudah menunggu dosen tersebut mengalami kerugian waktu dan sulit untuk menentukan jadwal bimbingan selanjutnya. Jika hal ini tidak segera diatasi, maka mahasiswa akan kesulitan untuk menyelesaikan laporan atau skripsi tepat waktu.

Penelitian sebelumnya [2] mengembangkan aplikasi menggunakan bahasa pemrograman Java dan *database* Firebase. Pada aplikasi ini, dosen dapat menambahkan jadwal bimbingan untuk mahasiswa pada tanggal dan waktu yang ditentukan. Agar lokasi dosen diketahui oleh mahasiswa, dosen mengaktifkan lokasi pada perangkatnya. *Google maps* berfungsi untuk menampilkan lokasi dosen dalam aplikasi.

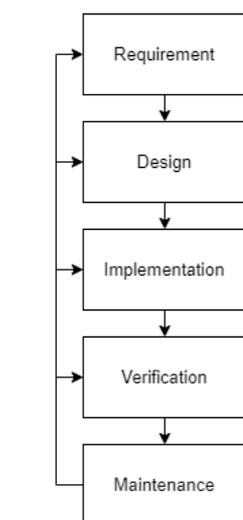
Pada halaman mahasiswa, jadwal bimbingan dari dosen muncul dan menampilkan status bimbingan serta lokasi dosen pada saat itu [2].

Penelitian lainnya [3] mengembangkan sistem ini menggunakan android studio dan *database* MySQL. Sistem ini memiliki dua tipe pengguna yaitu *client* dan pegawai *barber*. *Client* dan pegawai harus memiliki akun. Untuk melakukan pemesanan, *client* melakukan login. Setelah berhasil login akan diarahkan ke halaman utama. Pada halaman utama, disediakan beberapa gambar jenis model rambut yang akan dipilih oleh *client*. *Client* menekan salah satu model rambut dan akan muncul halaman lokasi *client* serta tombol untuk melakukan pemesanan. Setelah melakukan pemesanan, akan muncul halaman yang berisi informasi pesanan, fitur *chat*, dan tombol membatalkan pemesanan. Pada halaman utama pegawai *barber*, akan muncul daftar pelanggan yang melakukan pemesanan dengan menampilkan informasi lokasi pelanggan dan jenis model rambut [3].

Berdasarkan latar belakang dan penelitian terdahulu yang telah diuraikan, peneliti mengembangkan aplikasi berbasis android yang diharapkan dapat mengatur antrian bimbingan kp dan proyek akhir mahasiswa Politeknik Caltex Riau jurusan teknologi informasi menggunakan algoritma *priority queue* yaitu antrian berdasarkan prioritas [4] dan memanfaatkan *location-based services* (LBS) untuk mendeteksi titik lokasi mahasiswa dalam area *geofence* dosen [5].

## II. METODE PENELITIAN

Tahapan metode penelitian dimulai dari penelitian menggunakan salah satu model pengembangan perangkat lunak atau yang dikenal sebagai *Software Development Life Cycle* (SDLC) yaitu *waterfall*. Metode ini melakukan pendekatan secara sistematis. Oleh karena itu, setiap proses dilakukan secara berurutan dengan menunggu proses sebelumnya selesai [6].



Gambar 1. Metode *Waterfall*

Langkah pertama dalam pengembangan perangkat lunak adalah *requirement* yaitu peneliti merumuskan permasalahan, lalu membuat kuesioner untuk diberikan kepada mahasiswa dan dosen yang bersangkutan untuk mengumpulkan informasi terkait penelitian dan pengembangan aplikasi. Langkah kedua yaitu proses *design*, peneliti merancang tampilan untuk pengguna sesuai dengan kebutuhan. Langkah selanjutnya yaitu proses *implementation* dan *verification*, peneliti melakukan pengujian terhadap logika atau alur kode program, fungsional aplikasi, lalu pengujian kepada pengguna. Langkah terakhir yaitu peneliti melakukan perbaikan dan peningkatan kode program.

Selanjutnya peneliti melakukan penelitian menggunakan tiga metode pengujian perangkat lunak, yaitu:

### A. Usability Testing

*Usability* adalah kemampuan perangkat lunak agar dapat digunakan oleh pengguna untuk mencapai tujuan tertentu. *Usability* termasuk kedalam salah satu karakteristik ISO 9126-1. *Usability testing* dilakukan untuk menilai kualitas terhadap sebuah perangkat lunak dengan mengetahui apakah perangkat lunak yang dibangun sudah sesuai dengan kebutuhan pengguna [7]. Pengujian ini dilakukan kepada pengguna aplikasi, yaitu dosen dan mahasiswa Politeknik Caltex Riau Jurusan Teknologi Informasi. Subkarakteristik dari *usability* meliputi *understandability*, *learnability*, *operability*, dan *attractiveness*.

### B. Blackbox Testing

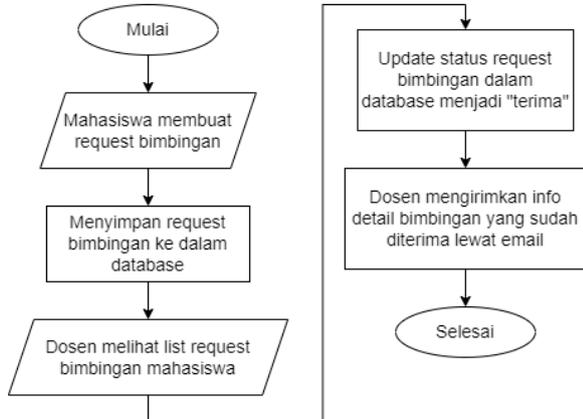
*Blackbox testing* merupakan pengujian terhadap kebutuhan fungsional sebuah perangkat lunak. Proses ini dilakukan tanpa mengetahui detail struktur internal dari sistem yang diuji. Pihak yang merencanakan perangkat lunak dapat menggunakan kondisi-kondisi inputan untuk memeriksa kebutuhan fungsional perangkat lunak secara keseluruhan. *Blackbox testing* dilakukan pada tahap akhir setelah pengembangan perangkat lunak [8].

### C. Whitebox Testing

*Whitebox testing* merupakan pengujian yang berfokus pada logika internal dan struktur kode. Teknik *whitebox testing* digunakan untuk debugging kode program yaitu menemukan kesalahan yang ada pada saat pengkodean. Sebelum melakukan perhitungan, terlebih dahulu membuat sebuah *flowchart* untuk menggambarkan logika program pada tahap perancangan dan *flowgraph* digunakan pada tahap pengujian yang berfokus pada alur program [9]. *Cyclomatic complexity* akan menghitung jumlah *path* program berdasarkan *edge* dan *node* yang ada pada *flowgraph* dan menentukan jumlah jalur independen.

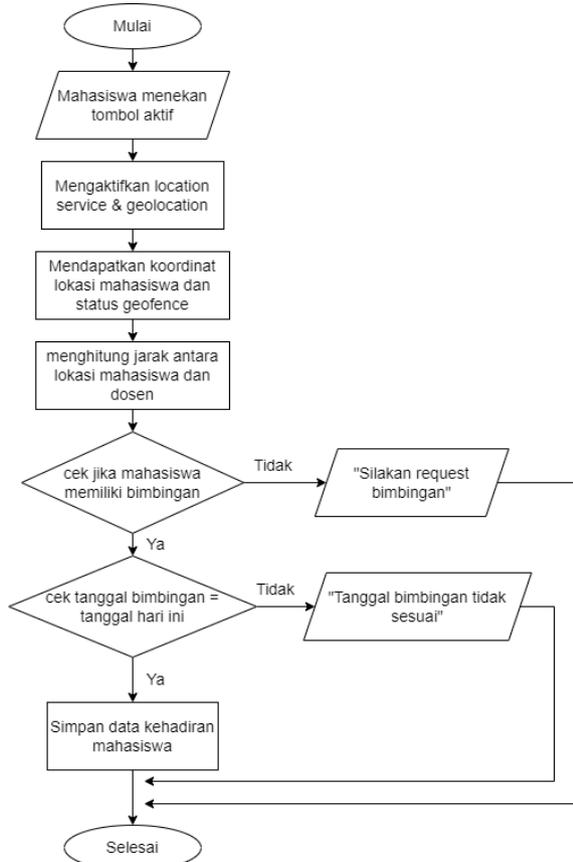
III. ANALISA DAN PERANCANGAN

A. Flowchart Setelah Menggunakan Aplikasi



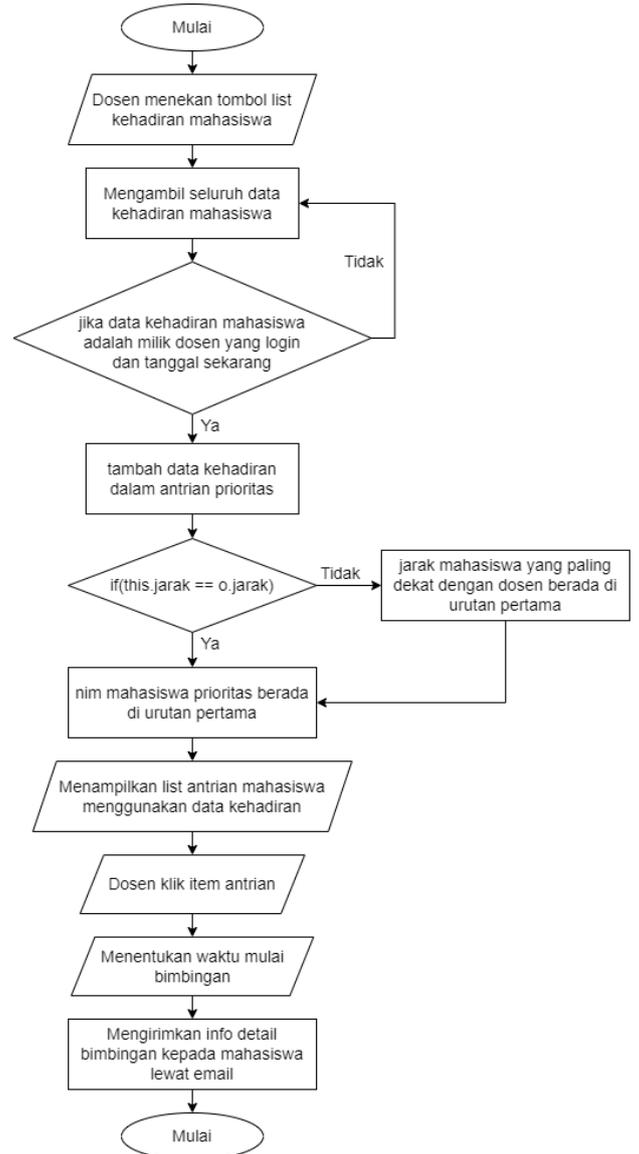
Gambar 2. Menambah dan Kelola Request Bimbingan

Gambar 2 menjelaskan bahwa mahasiswa menambahkan request bimbingan dengan memilih nama dosen, tanggal, dan jenis bimbingan yang akan dilakukan. Selanjutnya, dosen akan melakukan konfirmasi terhadap request bimbingan dengan menerima bimbingan. Setelah itu, dosen mengirimkan informasi berupa detail bimbingan kepada mahasiswa lewat email. Request bimbingan yang sudah dikonfirmasi akan ditampilkan pada halaman list jadwal bimbingan.



Gambar 3. Check-In Kehadiran oleh Mahasiswa

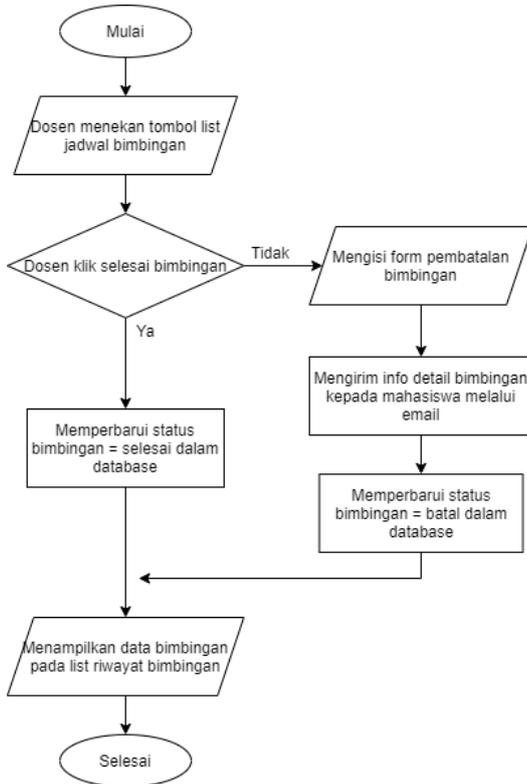
Gambar 3 menjelaskan alur mahasiswa melakukan kehadiran di area geofence dosen setelah melakukan request bimbingan. Mahasiswa dapat mengaktifkan location service dan geofence untuk mendapatkan titik koordinat lokasi mahasiswa agar dapat menghitung jarak antara mahasiswa dan dosen, serta mendeteksi lokasi mahasiswa didalam atau diluar area dosen. Kemudian dilakukan pengecekan data jika mahasiswa yang sedang login sudah memiliki jadwal bimbingan pada tanggal yang sama dengan hari ini maka akan ditambahkan kedalam database lalu ditampilkan melalui menu kehadiran mahasiswa.



Gambar 4. Antrian Bimbingan Berdasarkan Jarak Lokasi Mahasiswa

Selanjutnya pada Gambar 4 menjelaskan proses algoritma antrian priority queue. Sebelum menentukan elemen akan dilakukan pengambilan seluruh data mahasiswa yang sudah check in kehadirannya dengan syarat bimbingan itu dilakukan oleh dosen yang sedang login dan tanggal hari ini, lalu akan

diperiksa. Jika benar, maka akan dilakukan proses perbandingan untuk menentukan elemen prioritas. Jarak mahasiswa dan dosen yang paling dekat akan menjadi indikator prioritas. Namun, jika antara mahasiswa memiliki jarak yang sama, maka akan didahulukan mahasiswa dengan nim paling kecil. Lalu data tersebut akan ditampilkan pada list kehadiran mahasiswa. Selanjutnya dosen mulai mengatur waktu bimbingan pada mahasiswa yang berada di urutan elemen pertama dan mengirimkan informasi detail jadwal bimbingannya melalui *email*.



Gambar 5. Kelola Jadwal Bimbingan

Flowchart terakhir ditampilkan pada Gambar 5 yang menjelaskan alur dosen mengelola jadwal bimbingan yang sudah diatur waktunya. Setelah memberikan bimbingan, dosen dapat menekan tombol selesai, lalu akan memperbarui status bimbingan menjadi selesai, atau dosen dapat menekan tombol batal, lalu mengisi pesan pembatalan yang nantinya akan dikirimkan melalui *email* kepada mahasiswa bersangkutan. Setelah mengelola jadwal bimbingan, data ini akan ditampilkan pada halaman list riwayat bimbingan.

**B. Proses Antrian Prioritas Menggunakan Priority Queue**

Berikut ini merupakan proses dalam menentukan antrian dengan algoritma *priority queue*, yaitu:

**1) Indikator Antrian**

Salah satu *variable* penting untuk proses menentukan antrian adalah *variable comparator*. Indikator yang digunakan yaitu jarak dan nim

mahasiswa.

**2) Proses Menambahkan Elemen**

Elemen berupa *object* yang spesifik akan ditambahkan kedalam *priority queue* menggunakan perintah *add*. Elemen-elemen yang tertampung didalamnya akan diurutkan berdasarkan prioritas. Syarat sebuah data kehadiran mahasiswa akan ditambahkan adalah data yang memiliki id dosen sama dengan id dosen dalam data kehadiran dan tanggal yang sama dengan tanggal hari ini.

```

    Helper_LocJarak acc = new Helper_LocJarak(a,e,h,i,c,b,f,g,d);

    if (b.equals(nomornya) & i.equals(tanggal)) {
        pq.add(acc);
    }
    
```

**3) Proses Perbandingan Elemen**

Elemen antrian diurutkan setelah dibandingkan menggunakan *method compareTo*. Metode ini akan mengembalikan elemen prioritas (terkecil) yaitu jarak paling dekat atau nim paling kecil.

```

    @Override
    public int compareTo(Helper_LocJarak o) {
        if(this.getJarak() == o.getJarak()) {
            return this.getNomor() < o.getNomor() ? -1 : 1;
        } else
            return this.getJarak() < o.getJarak() ? -1 : 1;
    }
    
```

Berikut ini akan dijelaskan proses menentukan elemen prioritas menggunakan studi kasus antrian mahasiswa. Diasumsikan terdapat 4 orang mahasiswa yang akan melakukan bimbingan dengan data yang ditampilkan pada Tabel 1 berikut.

TABEL 1  
CONTOH DATA MAHASISWA BIMBINGAN

Nama	NIM	Jarak Mahasiswa dan Dosen (m)
Putra	1555301005	18
Maulana	1555301105	10
Alma	1755301105	10
Putri	2055301105	7

Proses 1: Melakukan pengambilan data mahasiswa dari *database* yang akan ditambahkan kedalam *priority queue*

Proses 2: Data “Putra” akan masuk pertama kali karena menjadi data pertama dalam *database*

1	2	3	4
Putra			

Proses 3: Setelah itu proses data kedua, melakukan perbandingan antara dua data mahasiswa, dan dicek apakah jarak “Putra” dan “Maulana”

bernilai sama, karena tidak maka akan dilanjutkan ke pernyataan selanjutnya (*else*) yaitu mendahulukan “Maulana” dengan jarak paling kecil

1	2	3	4
Maulana	Putra		

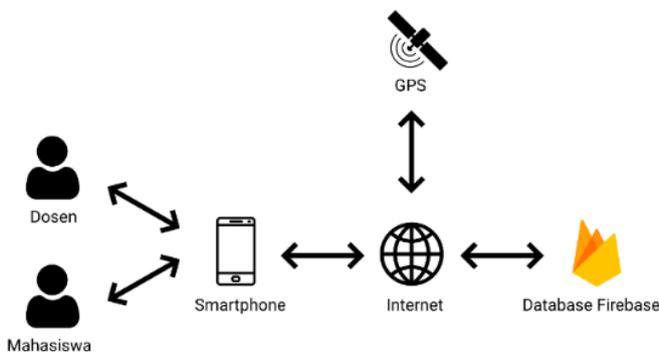
Proses 4: Setelah itu proses data ketiga, melakukan perbandingan antara dua data mahasiswa, dan dicek apakah jarak “Maulana” dan “Alma” bernilai sama, karena iya maka elemen yang didahulukan adalah “Maulana”, sebab nim 15 lebih kecil dari nim 17

1	2	3	4
Maulana	Putra	Alma	

Proses 5: Setelah itu proses data keempat, melakukan perbandingan antara dua data mahasiswa, dan dicek apakah jarak “Maulana” dan “Putri” bernilai sama, karena tidak, maka elemen yang didahulukan adalah “Putri” sebab jaraknya lebih dibandingkan jarak milik maulana. Oleh karena itu, elemen “Putri” pindah ke urutan pertama.

1	2	3	4
Putri	Maulana	Alma	Putra

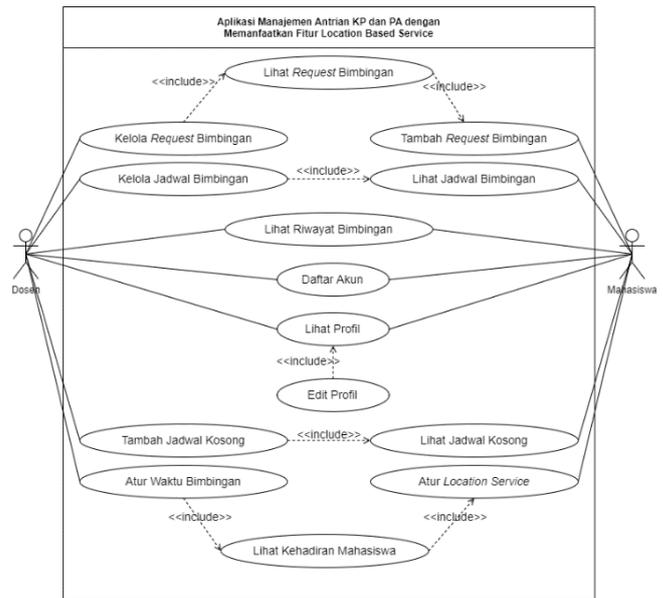
C. Arsitektur Aplikasi



Gambar 6. Arsitektur Aplikasi

Gambar 6 menjelaskan tentang arsitektur dalam aplikasi manajemen antrian bimbingan. Pengguna terdiri dari dosen dan mahasiswa Politeknik Caltex Riau jurusan teknologi informasi yang terlibat dalam proses bimbingan. Masing-masing pengguna harus menghubungkan *smartphone* nya dengan internet, kemudian mengaktifkan GPS untuk mendapatkan koordinat lokasi pengguna berupa *longitude* dan *latitude* yang akan disimpan ke dalam *database*. Internet berfungsi sebagai jembatan pengiriman data dari *smartphone* ke *database*. Sedangkan Firebase berfungsi sebagai *Backend as Service* dan tempat penyimpanan data [10].

D. Use Case Diagram



Gambar 7 Use Case Diagram

Gambar 7 menjelaskan aksi yang dapat dilakukan oleh aktor dosen dan mahasiswa. Setiap *use case* dapat berfungsi setelah kedua aktor melakukan *login*. Jika aktor belum memiliki akun, maka harus melakukan pendaftaran. Mahasiswa dapat melakukan *request* bimbingan kepada dosen. Lalu, dosen dapat mengelola *request* bimbingan mahasiswa dengan konfirmasi terima bimbingan. Data bimbingan akan masuk kedalam list jadwal bimbingan dengan tanggal yang sudah ditentukan.

Selanjutnya, mahasiswa melakukan konfirmasi kehadiran pada tanggal bimbingannya dengan cara mengaktifkan *location service* untuk mendapatkan koordinat lokasi terkini dan *geofence* untuk mendeteksi apakah koordinat tersebut berada dalam area *geofence* dosen atau tidak. Setelah berhasil mendapatkan titik lokasi, maka akan disimpan ke dalam *database* dan dihitung jarak antara titik dosen dan titik mahasiswa saat ini. Jika mahasiswa hadir, maka akan diurutkan berdasarkan prioritasnya yang dapat dilihat pada halaman list kehadiran mahasiswa oleh dosen. Lalu dosen mengatur waktu bimbingan untuk mahasiswa tersebut dan bisa memberikan bimbingan. Dosen dapat mengelola jadwal bimbingan dengan keterangan selesai atau batal. Setelah diproses, jadwal bimbingan ini akan ditampilkan pada halaman list riwayat bimbingan.

Dosen dapat menambahkan informasi jadwal kosong kepada mahasiswa melalui kalender yang disediakan, lalu mahasiswa dapat mengklik info tersebut dan akan menampilkan daftar bimbingan yang sedang diproses pada tanggal tersebut dan oleh dosen tertentu. Terakhir, kedua aktor juga dapat melihat dan mengedit data profil.

#### IV. HASIL DAN PEMBAHASAN

##### A. Implementasi Aplikasi

Hasil dari pengembangan ini adalah sebuah aplikasi *mobile* berbasis android yang dapat digunakan oleh dua tipe pengguna yaitu dosen dan mahasiswa.

##### 1) Halaman Pendaftaran dan Login

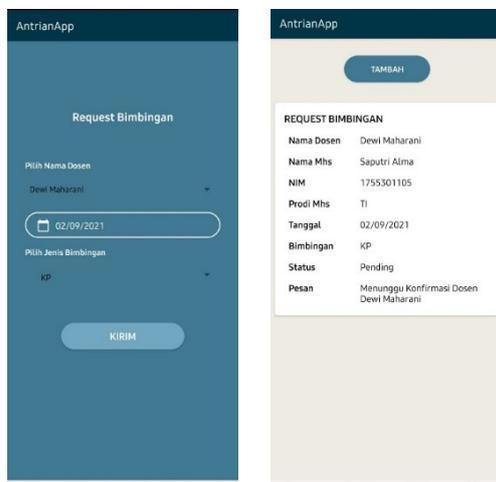
Saat melakukan pendaftaran akun, pengguna akan mengisi data nama, NIP / NIM, program studi (TI, SI, TK), nomor telp, alamat email, *password*, dan *level user* yang terdiri dari Mahasiswa dan Dosen. Setelah selesai melakukan pendaftaran, akan di alihkan ke halaman *Login*. Lalu pengguna dapat menginputkan *nim/nip* dan *password*.

##### 2) Halaman Utama

Pada halaman ini terdapat menu-menu yang dapat digunakan oleh dosen atau mahasiswa. Pada sudut kanan atas halaman ini terdapat tombol untuk *logout*. Setelah *login* akan ditampilkan nama dan *nip/nim* pengguna.

##### 3) Halaman Request Bimbingan Mahasiswa

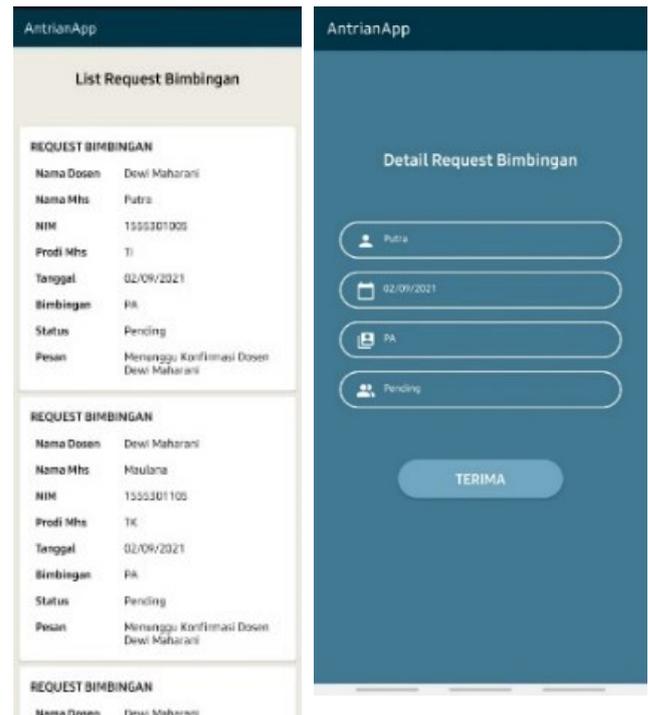
Sebelum mendapatkan jadwal bimbingan, mahasiswa melakukan *request* bimbingan agar dapat diproses oleh dosen. Mahasiswa menekan tombol tambah, lalu mengisi data yaitu memilih nama dosen, mengisi tanggal, dan memilih jenis bimbingan. Setelah menekan tombol kirim, maka akan tampil data request bimbingan yang memiliki status “Pending” seperti yang ditampilkan pada Gambar 8.



Gambar 8. Halaman Request Bimbingan

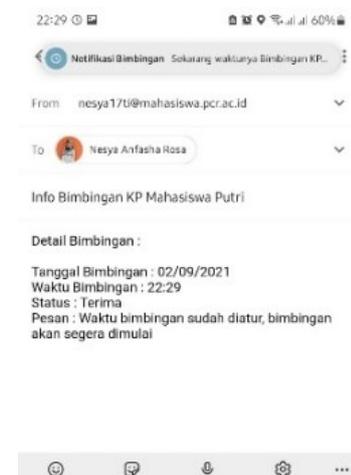
##### 4) Halaman List Request Bimbingan Dosen

Setelah mahasiswa melakukan request bimbingan, halaman ini akan menampilkan data seluruh mahasiswa yang sudah menambahkan request bimbingan pada dosen tersebut. Data yang akan tampil adalah data bimbingan dengan status “Pending” dan waktu belum diatur seperti yang ditampilkan pada Gambar 9.



Gambar 9. Halaman Request Bimbingan

Dosen dapat menekan item request bimbingan untuk dikonfirmasi terima agar masuk kedalam list jadwal bimbingan, Lalu, dosen dapat mengirimkan informasi detail jadwal bimbingan kepada mahasiswa lewat email seperti yang ditunjukkan oleh Gambar 10.

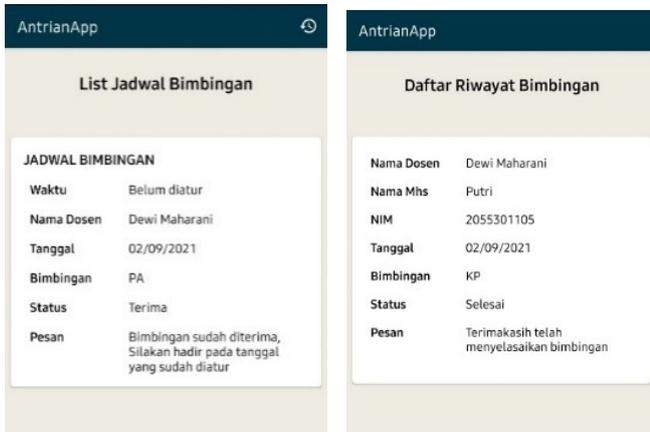


Gambar 10. Pengiriman Info Detail Bimbingan di Email

##### 5) Halaman List Jadwal dan Riwayat Bimbingan Mahasiswa

Pada halaman ini terdapat data jadwal bimbingan mahasiswa itu sendiri yang sudah diterima oleh dosen bersangkutan dengan status “Terima”, serta menampilkan tanggal. Pada pojok kanan atas terdapat menu untuk melihat

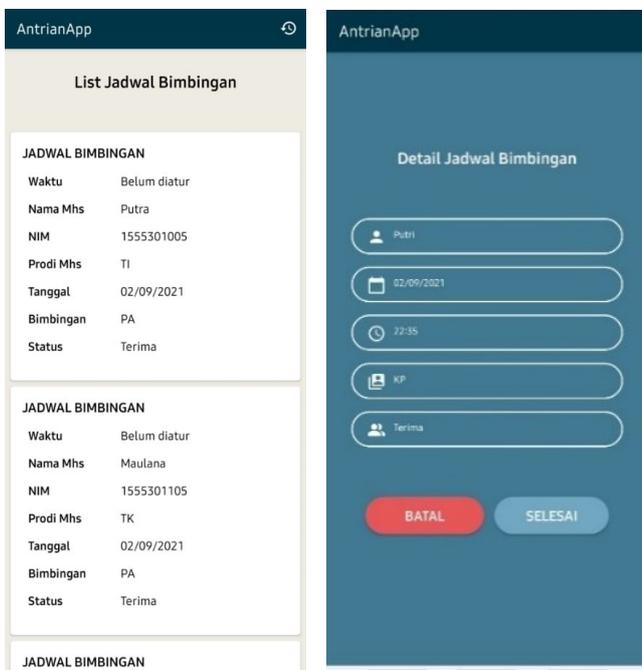
“Daftar Riwayat Bimbingan” yaitu jadwal bimbingan mahasiswa yang memiliki status Batal atau Selesai seperti yang ditunjukkan pada Gambar 11.



Gambar 11. Halaman List Jadwal dan Riwayat Bimbingan

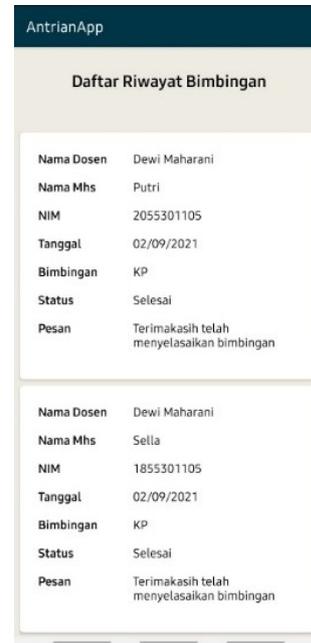
6) Halaman List Jadwal dan Riwayat Bimbingan Dosen

Halaman ini menampilkan jadwal bimbingan seluruh mahasiswa yang sudah diterima oleh dosen bersangkutan dengan status “Terima”. Dosen dapat mengelola jadwal bimbingan mahasiswa dengan menekan item jadwal bimbingan untuk diarahkan ke halaman detail jadwal bimbingan. Jika dosen ingin membatalkan bimbingan, maka dosen harus mengisi pesan pembatalan. Tetapi jika dosen telah menyelesaikan bimbingan maka dosen dapat menekan tombol selesai seperti pada Gambar 12.



Gambar 12. Halaman List Jadwal Bimbingan Dosen

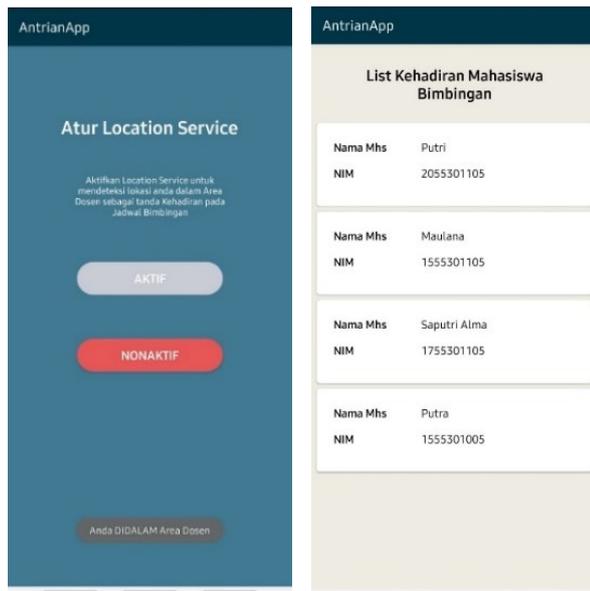
Pada pojok kanan atas terdapat menu untuk melihat “Daftar Riwayat Bimbingan” yaitu jadwal bimbingan seluruh mahasiswa dengan dosen tersebut dengan status Batal atau Selesai seperti pada Gambar 13.



Gambar 13. Halaman Riwayat Bimbingan Dosen

7) Halaman Kehadiran Mahasiswa

Awalnya mahasiswa melakukan *check in* kehadiran pada tanggal bimbingan menggunakan *location service* agar mendapatkan lokasi akurat dan mengecek posisi mahasiswa apakah berada didalam *geofence* dosen atau tidak, serta menghitung jarak antara lokasi mahasiswa dan dosen. Untuk menjalankan *service* dan mendeteksi lokasi dalam *geofence* dosen, mahasiswa dapat menekan tombol aktif, sedangkan untuk mematikan *service* dan *geofence*, mahasiswa dapat menekan tombol nonaktif, seperti yang ditunjukkan pada Gambar 14. Pada halaman ini, kehadiran mahasiswa akan diurutkan berdasarkan prioritas jarak paling dekat dengan area dosen menggunakan algoritma *priority queue*, yang ditampilkan pada Gambar 14. Untuk melanjutkan proses bimbingan, dosen dapat menekan item antrian mahasiswa yang pertama untuk menentukan waktu bimbingan. Setelah itu akan diatur notifikasi pengingat bimbingan kepada dosen dan mengirim informasi detail bimbingan kepada mahasiswa. Data jadwal bimbingan milik mahasiswa yang dipilih akan diperbarui untuk menampilkan informasi waktu bimbingan.



Gambar 14. Kehadiran Mahasiswa

### B. Analisa Implementasi Priority Queue

Antrian prioritas diimplementasikan pada saat mahasiswa melakukan kehadiran bimbingan agar mendapatkan kepastian waktu dari dosen. Pada halaman list kehadiran mahasiswa akan menampilkan antrian mahasiswa berdasarkan prioritas jarak terdekat dengan dosen, dan jika terdapat mahasiswa dengan jarak yang sama, maka akan diprioritasnya berdasarkan nim mahasiswa paling kecil atau generasi tertua. Setelah mengimplementasikan antrian, terlihat bahwa urutan yang ada dalam priority queue tidak selalu berurut secara natural. Karena, menurut Cherymy [11] priority queue hanya menjamin elemen antrian pertama atau head bernilai kecil atau besar bergantung pada perbandingan yang sudah ditetapkan.

Dalam java, priority queue adalah sebuah heap. Penentuan urutan head antrian prioritas terbagi kedalam dua jenis, yaitu secara ascending dan descending [12]. Jika ingin menampilkannya dengan urutan antrian yang benar, maka dapat menggunakan metode poll. Namun, dalam penggunaannya, metode ini berfungsi untuk menghapus elemen yang paling depan hingga elemen dalam antrian kosong. Oleh karena itu, method ini hanya berfungsi jika ingin mencetak elemennya saja. Sedangkan, dalam aplikasi ini, elemen ingin disimpan ke dalam database dan ditampilkan.

### C. Analisa Pengujian Dengan Whitebox Testing

Berdasarkan hasil pengujian whitebox terhadap algoritma priority queue didapatkan edge sebanyak 15 dan node sebanyak 15, lalu melakukan perhitungan cyclomatic complexity untuk mendapatkan nilai V(G) yang hasilnya adalah 2. Dapat ditarik kesimpulan bahwa kode program terstruktur dan sudah ditulis dengan baik.

Jalur independen dibutuhkan untuk melakukan eksekusi pada alur kode program menggunakan test case. Pengujian terhadap 2 test case memberikan hasil yang diharapkan dan sebenarnya yaitu sesuai dengan tingkat keberhasilan eksekusi masing-masing jalur sebesar 100%. Oleh karena itu, kesimpulan dari pengujian ini adalah kondisi yang diharapkan berhasil terpenuhi dan kode program algoritma priority queue berhasil dieksekusi.

### D. Analisa Pengujian Dengan Blacbox Testing

Tujuan dari blackbox testing adalah untuk melakukan pengujian terhadap kebutuhan fungsional aplikasi antrian bimbingan. Pengujian ini dilakukan terhadap 23 testcase pada sisi mahasiswa dan 24 testcase pada sisi dosen. Setelah melakukan pengujian, didapatkan hasil bahwa 100% seluruh testcase milik mahasiswa dan dosen berhasil berfungsi sesuai yang diharapkan.

### E. Analisa Pengujian Dengan Usability Testing

Berdasarkan Tabel 4.1, didapatkan hasil analisa kuesioner pengujian usability testing setelah dihitung menggunakan skala likert [13]. Secara keseluruhan, setiap pernyataan mendapatkan pernyataan "Setuju" dengan persentase nilainya yang berada dalam range 80%-100%. Dapat ditarik kesimpulan bahwa mahasiswa dan dosen setuju bahwa aplikasi antrian bimbingan sesuai dengan pernyataan yang sudah ada, dengan rata-rata persentase sebesar 85,4%.

TABEL 2  
HASIL ANALISA KUESIONER

No.	Pernyataan	Persentase	Nilai
1.	Informasi yang diberikan aplikasi mudah untuk dipahami	85 %	Setuju
2.	Aplikasi mudah untuk digunakan	87 %	Setuju
3.	Aplikasi mudah untuk dipelajari	85 %	Setuju
4.	Aplikasi memiliki tampilan yang menarik	82 %	Setuju
5.	Fitur yang ada didalam aplikasi memenuhi kebutuhan	86 %	Setuju
6.	Aplikasi berjalan sesuai dengan apa yang saya harapkan	85 %	Setuju
7.	Aplikasi menerapkan penggunaan location-based service dengan tepat (GPS dan geofence)	88 %	Setuju
8.	Aplikasi dapat mengatur antrian bimbingan dengan baik	84 %	Setuju
9.	Aplikasi dapat diterapkan di Politeknik Caltex Riau	87 %	Setuju
Rata-Rata		85,4 %	Setuju

### F. Analisa Pengujian Titik Lokasi Mahasiswa dalam Geofence

Pengujian titik lokasi mahasiswa dilakukan pada 3 lokasi yaitu didepan ruang dosen TI, SI dan TK yang berada di lantai 3 gedung kuliah PCR. Pengujian dilakukan dengan menetapkan 4 jenis ukuran radius area geofence dosen yaitu

30, 40, 50, dan 60 (meter). Berdasarkan hasil pengujian bahwa pada radius 30-40 m sekitar dosen, lokasi mahasiswa tidak terdeteksi seluruhnya. Kemudian pada radius 50-60 m sekitar dosen, lokasi mahasiswa berhasil terdeteksi seluruhnya. Menurut penjelasan dari Android Developers [5] [14], jika ingin mendapatkan hasil yang baik, radius geofence harus diatur sebesar 100-150 meter. Jika pengguna menggunakan Wi-Fi, akurasi lokasinya adalah 20-50 meter. Oleh karena itu, lokasi mahasiswa berhasil terdeteksi seluruhnya dalam geofence jika radiusnya mulai dari 50 meter.

## V. KESIMPULAN

Setelah mengerjakan proyek akhir, peneliti dapat mengambil kesimpulan bahwa berdasarkan hasil pengujian perangkat lunak menggunakan teknik *usability testing*, menunjukkan hasil bahwa pengguna aplikasi (responden) setuju terhadap pernyataan yang terdapat dalam kuesioner dengan rata-rata persentase sebesar 85,4%. Pada pengujian *blackbox* dan *whitebox* menunjukkan bahwa keberhasilan eksekusi kode program dan fungsional aplikasi adalah sebesar 100%. Untuk pengembangan selanjutnya diperlukan peningkatan kode program dan penyesuaian dengan teknologi yang paling terbaru agar seluruh jenis perangkat android dapat mendukung penggunaan aplikasi ini.

## UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada pihak yang telah banyak memberikan bantuan dan dukungan secara langsung maupun tidak langsung kepada Politeknik Caltex Riau, Politeknik Negeri Batam, serta pihak-pihak lainnya yang terkait.

## DAFTAR PUSTAKA

- [1] "Manual Prosedur (MP) Proyek Akhir Mahasiswa Politeknik Caltex Riau," 2018.
- [2] R. A. Fauji and A. Rahmatulloh, 'Implementation Of Global Positioning System As A Location Monitoring Media For Final Project Guidelines On Android-Based Applications', *J. Transform.*, vol. 16, no. 2, Art. no. 2, Jan. 2019, doi: 10.26623/transformatika.v16i2.1071.
- [3] R. Bagus Bratta, 'Implementasi Algoritma Queue Untuk Sistem Antrian Booking Barbershop Berbasis Android', skripsi, IIB DARMAJAYA, 2018. Accessed: Jun. 23, 2022. [Online]. Available: <http://repo.darmajaya.ac.id/1276/>
- [4] A. D. "PriorityQueue," 27 12 2019. [Online]. Available: <https://developer.android.com/reference/java/util/PriorityQueue>. [Accessed 16 12 2020].
- [5] A. D. "Create and Monitor Geofences," 16 8 2021. [Online]. Available: <https://developer.android.com/training/location/geofencing>. [Accessed 18 1 2021].
- [6] A. A. Wahid, 'Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi', *J Ilmu-Ilmu Inf. Dan Manaj STMIK No Novemb.*, pp. 1–5, 2020.
- [7] K. Naik and P. Tripathy, *Software Testing and Quality Assurance Theory and Practice*, A John Wiley & Sons, Inc., 2008.
- [8] A. Putri, 'Aplikasi Pencarian Hotel di Pekanbaru Berdasarkan Informasi Promo dan Diskon Berbasis Location Based Service (LBS)', *J. Aksara Komput. Terap.*, vol. 7, no. 2, Art. no. 2, 2018, Accessed: Jun. 23, 2022. [Online]. Available: <https://jurnal.pcr.ac.id/index.php/jakt/article/view/1982>.
- [9] D. D. A. Suprpto, F. I. Fitri and N. Hayati, "Pengembangan Aplikasi Sistem Informasi Smart Register Online Berbasis Android Menggunakan Algoritma BruteForce," *Edumatic : Jurnal Pendidikan Indonesia*, vol. 4, no. 1, pp. 47-56, 1 Juni 2020.
- [10] M. Michelle, 'Rancang Bangun Aplikasi Context-Aware "Family Tracking" Berbasis Mobile pada Platform iOS', *J. Aksara Komput. Terap.*, vol. 7, no. 2, Art. no. 2, 2018, Accessed: Jun. 23, 2022. [Online]. Available: <https://jurnal.pcr.ac.id/index.php/jakt/article/view/1992>.
- [11] M. Cheramy, "Priority Queue is not Maintaining Sorting Order," 10 August 2018. [Online]. Available: <https://stackoverflow.com/questions/51787067/priority-queue-is-not-maintaining-sorting-order>.
- [12] A. Sridhar, "Priority Queues in Java Explained with Examples," 6 Juli 2020. [Online]. Available: <https://www.freecodecamp.org/news/priority-queue-implementation-in-java/>.
- [13] A. Novriansyah and D. E. Kurniawan, 'Pengembangan Aplikasi Pemasaran Ikan Untuk UMKM dan Nelayan Batam dengan Pendekatan User Centered Design (UCD) dan Usability Testing', *J. Appl. Comput. Sci. Technol.*, vol. 3, no. 1, Art. no. 1, Jun. 2022, doi: 10.52158/jacost.v3i1.310.
- [14] D. E. Kurniawan and A. Algazali, *Mobile Programming: Praktik Membuat Aplikasi Android*. Media Sains Indonesia, 2022.