

**PENGEMBANGAN APLIKASI PERANGKAT BERGERAK:
PENCARI KONSELOR PSIKOLOGI TERDEKAT BERBASIS
LOKASI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Tifo Audi Alif Putra
NIM: 165150200111168



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PERSETUJUAN

**PENGEMBANGAN APLIKASI PERANGKAT BERGERAK: PENCARI KONSELOR
PSIKOLOGI TERDEKAT BERBASIS LOKASI**

SKRIPSI

**Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer**

**Disusun Oleh :
Tifo Audi Alif Putra
NIM: 165150200111168**

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing 2

Agi Putra Kharisma, S.T., M.T.

NIK: 201304 860430 1 001

Randy Cahya Wihandika, S.ST., M.Kom.

NIK: 201405 880206 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Januari 2020

Tifo Audi Alif Putra

NIM: 165150200111168

PRAKATA

Segala puji bagi Allah SWT karena atas berkah limpahan rahmat dan hidayah-Nya lah penulis dapat menyelesaikan skripsi yang berjudul “Pengembangan Aplikasi Perangkat Bergerak: Aplikasi Pencari Konselor Psikologi Terdekat Berbasis Lokasi” ini. Dalam kesempatan ini, penulis ingin memberikan ucapan terima kasih kepada seluruh pihak yang telah membantu penulis dalam proses penyusunan skripsi ini, diantaranya:

1. Kedua orang tua penulis, yaitu Bapak Roediono dan Ibu Titik Hastuti yang beserta seluruh keluarga besar yang terus memberikan dukungan dan motivasi serta doa yang tidak kunjung putus.
2. Bapak Agi Putra Kharisma, S.T., M.T. selaku dosen pembimbing I yang telah meluangkan waktu dan tenaganya untuk memberi masukan, ilmu, dan saran yang bermanfaat dalam proses penyusunan skripsi ini.
3. Bapak Randy Cahya Wihandika, S.ST., M.Kom. selaku dosen pembimbing II yang telah meluangkan waktu dan tenaganya untuk memberi masukan, ilmu, dan saran yang bermanfaat dalam proses penyusunan skripsi ini.
4. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., Bapak Ir. Heru Nurwasito, M.Kom., Bapak Suprpto, S.T., M.T., dan Bapak Edy Santoso, S.Si., M.Kom. selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. dan Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Ketua Jurusan Teknik Informatika dan Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya dalam mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
7. Ibu Dita Rachmayani, S.Psi., M.A. selaku dosen psikologi dan narasumber yang telah meluangkan waktu dan tenaganya untuk wawancara menggali kebutuhan kesehatan mental di Indonesia yang akan menjadi acuan dalam penelitian.
8. Seluruh teman dan sahabat penulis dalam komunitas BCC Filkom UB yang secara tidak langsung memberikan ilmu serta motivasi dalam penyusunan skripsi ini.
9. Semua pihak yang tidak semuanya bisa dituliskan disini yang terlibat secara langsung maupun tidak langsung dalam penyusunan skripsi ini.

Penulis sadar masih banyak kekurangan dan kesalahan yang terdapat dalam penyusunan skripsi ini baik dari sisi teknis penelitian maupun penyajian materi. Maka dari itu, penulis mengharapkan kritik dan saran yang membangun dari

pembaca. Semoga karya tulis ini bermanfaat dan dapat memberikan manfaat bagi pihak yang membutuhkan.

Malang, 15 Desember 2019

Penulis

tifoaudy@gmail.com

ABSTRAK

Tifo Audi Alif Putra, Pengembangan Aplikasi Perangkat Bergerak: Pencari Konselor Psikologi Terdekat Berbasis Lokasi

Pembimbing: Agi Putra Kharisma, S.T., M.T. dan Randy Cahya Wihandika, S.ST., M.Kom.

Kesehatan mental saat ini menjadi permasalahan yang tidak bisa diremehkan. Bahkan kesehatan mental sekarang menjadi masalah yang dihadapi banyak negara di dunia, tak terkecuali Indonesia. Depresi menjadi penyebab utama dalam permasalahan kesehatan mental. Depresi memengaruhi 4,4% populasi dunia dan 5% populasi di Indonesia. Selain jumlah penderita depresi yang tinggi, masalah lain pada Indonesia adalah kurangnya jumlah ketersediaan tenaga profesional kesehatan mental seperti konselor dan psikiater. Maka dari itu, pemanfaatan sistem berbasis *mobile* dapat dilakukan untuk menghubungkan orang yang memiliki permasalahan dengan konselor terdekat untuk mendapatkan pertolongan pertama dalam masalahnya. Implementasi pengembangan aplikasi perangkat bergerak pada penelitian ini dikembangkan pada platform *iOS* dengan menggunakan bahasa pemrograman *Swift*. Metode penelitian yang digunakan adalah *iterative waterfall model* yang terdiri dari analisis kebutuhan, perancangan sistem, implementasi sistem, dan pengujian. Pengujian pada penelitian ini terdiri dari *blackbox testing* dan *usability testing*. Pada *blackbox testing* peneliti merancang *testcase scenario* untuk menilai perilaku sistem yang benar sedangkan untuk mengukur tingkat *usability* dari aplikasi yang dikembangkan menggunakan instrumen SUPR-Qm. Hasil pengujian pada *blackbox testing* menyatakan 100% valid dari semua *testcase* yang dirancang. Untuk hasil pengujian *usability* diperoleh nilai 69,25% untuk aplikasi pasien dan 74,75% untuk aplikasi konselor, maka jika dikonversikan kedalam *usability rating scale* menghasilkan kategori *ok* untuk aplikasi pasien dan *good* untuk aplikasi konselor.

Kata kunci: kesehatan mental, depresi, *iOS*, *Swift*, *waterfall*, aplikasi perangkat bergerak.

ABSTRACT

Tifo Audi Alif Putra, Mobile Application Development: Location-Based Nearest Psychology Counselor Seekers

Supervisors: Agi Putra Kharisma, S.T., M.T. dan Randy Cahya Wihandika, S.ST., M.Kom.

Mental health is now a problem that cannot be underestimated. Even mental health is now a problem faced by many countries in the world, including Indonesia. Depression is a major cause of mental health problems. Depression affects 4.4% of the world's population and 5% of the population in Indonesia. In addition to the high number of people with depression, another problem in Indonesia is the lack of available mental health professionals such as counselors and psychiatrists. Therefore, the use of mobile-based systems can be done to connect people who have problems with the nearest counselor to get first aid in the problem. The implementation of mobile application development in this study was developed on the iOS platform using the Swift programming language. The research method used is an iterative waterfall model consisting of needs analysis, system design, system implementation, and testing. Testing in this study consisted of blackbox testing and usability testing. In the blackbox testing the researchers designed a testcase scenario to assess the correct system behavior while to measure the usability level of the application developed using the SUPR-Qm instrument. The test results on Blacbox Testing stated 100% valid of all designed testcases. For the usability test results obtained 69.25% for patient applications and 74.75% for counselor applications, then if converted to usability rating scale produces an ok category for patient applications and good for counselor applications.

keywords : mental health, depression, iOS, Swift, waterfall, mobile-based system.

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xiv
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat	2
1.5 Batasan Masalah.....	2
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN.....	4
2.1 Kajian Pustaka.....	4
2.2 <i>Mobile Application</i>	4
2.2.1 iOS.....	5
2.3 <i>Iterative Waterfall Model</i>	6
2.4 <i>Location-Based Service</i>	7
2.5 <i>Best Practices</i>	7
2.5.1 Swift	7
2.5.2 MVVM (Model-View-ViewModel).....	7
2.5.3 MapKit.....	8
2.5.4 Firebase.....	8
2.6 Pengujian Perangkat Lunak	8
2.6.1 <i>BlackBox Testing</i>	8
2.6.2 <i>Usability Testing</i>	9
2.6.3 <i>Skala Likert</i>	9
2.7 <i>Standardized User Experience Percentile Rank Questionnaire for Mobile Apps (SUPR-Qm)</i>	9
BAB 3 METODOLOGI	11

3.1 Studi Literatur	11
3.2 Analisis Kebutuhan.....	12
3.3 Perancangan dan Implementasi	12
3.3.1 Perancangan.....	12
3.3.2 Implementasi.....	13
3.4 Pengujian	13
3.4.1 Analisis Hasil Pengujian.....	13
3.5 Kesimpulan	13
BAB 4 ANALISIS KEBUTUHAN	14
4.1 Gambaran Umum Sistem	14
4.1.1 Deskripsi Umum Sistem	14
4.1.2 Lingkungan Sistem.....	14
4.2 Analisis Kebutuhan Perangkat Lunak.....	15
4.2.1 Identifikasi Aktor	15
4.2.2 Kebutuhan Fungsional Sistem.....	15
4.2.3 Kebutuhan Non-Fungsional Sistem	16
4.2.4 Usecase Diagram	17
4.2.5 Usecase Scenario	18
BAB 5 PERANCANGAN DAN IMPLEMENTASI	26
5.1 Arsitektur Sistem	26
5.2 Sequence Diagram	27
5.2.1 <i>Sequence Diagram</i> - Login	27
5.2.2 <i>Sequence Diagram</i> - Register	28
5.2.3 <i>Sequence Diagram</i> - Melihat Lokasi Konselor Dalam Peta	29
5.2.4 <i>Sequence Diagram</i> - Melihat Daftar Konselor	29
5.2.5 <i>Sequence Diagram</i> - Melihat Profile Konselor	30
5.2.6 <i>Sequence Diagram</i> - Mengirim Permintaan Konseling.....	31
5.2.7 <i>Sequence Diagram</i> - Melakukan Konseling.....	31
5.2.8 <i>Sequence Diagram</i> - Memberi Rating ke Konselor	32
5.2.9 <i>Sequence Diagram</i> - Melihat Riwayat Konseling	33
5.2.10 <i>Sequence Diagram</i> - Membuat <i>Appointment</i>	34
5.2.11 <i>Sequence Diagram</i> - Melihat Daftar Appointment Dengan Konselor	34
5.2.12 <i>Sequence Diagram</i> - Melihat Daftar Permintaan Konseling.....	35

5.2.13 <i>Sequence Diagram</i> - Melihat Detail Profil Pasien.....	36
5.2.14 <i>Sequence Diagram</i> - Merespon Permintaan Konseling.....	37
5.2.15 Melihat Daftar Appointment Dari Pasien	38
5.3 Class Diagram.....	39
5.4 Perancangan Struktur Data	43
5.5 Perancangan Algoritme.....	45
5.6 Perancangan Antarmuka.....	46
5.6.1 Screenflow - Aplikasi Sisi Pasien.....	47
5.6.2 Screenflow - Aplikasi Sisi Konselor	48
5.7 Implementasi Perangkat Lunak	49
5.7.1 Spesifikasi Lingkungan Sistem.....	49
5.7.2 Batasan Implementasi	50
5.7.3 Implementasi Struktur Data.....	50
5.7.4 Implementasi Kode Program	52
5.7.5 Implementasi Antarmuka	55
BAB 6 PENGUJIAN.....	58
6.1 <i>Blackbox Testing</i>	58
6.1.1 Login.....	58
6.1.2 Register	58
6.1.3 Melakukan Konseling.....	59
6.1.4 Membuat Appointment.....	60
6.2 Pengujian Usabilitas.....	60
6.2.1 Skenario Pengujian	62
6.2.2 Hasil Pengujian Usabilitas	63
6.2.3 Analisis Hasil Pengujian.....	67
BAB 7 PENUTUP	69
7.1 Kesimpulan	69
7.2 Saran.....	70
DAFTAR REFERENSI.....	71
LAMPIRAN A HASIL EXPERT INTERVIEW	74
LAMPIRAN B HASIL PENGUJIAN USABILITAS	76

DAFTAR TABEL

Tabel 4.1 Daftar Aktor.....	15
Tabel 4.2 Kebutuhan Fungsional - Pasien	15
Tabel 4.3 Kebutuhan Fungsional - Konselor	16
Tabel 4.4 Usecase Scenario - Login.....	19
Tabel 4.5 Usecase Scenario - Register	19
Tabel 4.6 Usecase Scenario - Melihat lokasi konselor dalam peta	20
Tabel 4.7 Usecase Scenario - Melihat daftar konselor	20
Tabel 4.8 Usecase Scenario - Melihat profile konselor	21
Tabel 4.9 Usecase Scenario - Mengirim permintaan konseling.....	21
Tabel 4.10 Usecase Scenario - Melakukan konseling.....	21
Tabel 4.11 Usecase Scenario - Memberikan rating kepada konselor	22
Tabel 4.12 Usecase Scenario - Membuat Appointment.....	22
Tabel 4.13 Usecase Scenario - Melihat Riwayat Konseling.....	23
Tabel 4.14 Usecase Scenario - Melihat Daftar Appointment Dengan Konselor ...	23
Tabel 4.15 Usecase Scenario - Melihat daftar permintaan konseling.....	23
Tabel 4.16 Usecase Scenario - Melihat detail profil user	24
Tabel 4.17 Usecase Scenario - Melihat Daftar <i>Appointment</i> Dari Pasien	24
Tabel 4.18 Usecase Scenario - Merespon permintaan konseling	25
Tabel 5.1 Penjelasan Class Diagram Pasien	40
Tabel 5.2 Penjelasan Class Diagram Konselor.....	42
Tabel 5.3 Perancangan Data Patient	43
Tabel 5.4 Perancangan Data Konselor.....	43
Tabel 5.5 Perancangan Data Order	44
Tabel 5.6 Perancangan Data Appointment.....	44
Tabel 5.7 Perancangan Data ChatRoom	44
Tabel 5.8 Perancangan Algoritme - Memperoleh Data Konselor Terdekat	45
Tabel 5.9 Perancangan Algoritme - Mengirim Permintaan Konseling	46
Tabel 5.10 Perancangan Algoritme - Membuat Appointment	46
Tabel 5.11 Penjelasan Screenflow Pasien.....	48
Tabel 5.12 Penjelasan Screenflow Konselor	49
Tabel 5.13 Spesifikasi Perangkat Keras.....	49
Tabel 5.14 Spesifikasi Perangkat Lunak	50
Tabel 6.1 <i>Blacbox testing</i> - Login.....	58
Tabel 6.2 <i>Blacbox testing</i> - Register	58

Tabel 6.3 <i>Blacbox testing</i> - Melakukan Konseling.....	59
Tabel 6.4 <i>Blacbox testing</i> - Membuat Appointment	60
Tabel 6.5 Skala Likert	61
Tabel 6.6 Daftar Pertanyaan SUPER-Qm	61
Tabel 6.7 Skenario Pengujian - Aplikasi Sisi Pasien	62
Tabel 6.8 Skenario Pengujian - Aplikasi Sisi Konselor.....	63
Tabel 6.9 Hasil Pengujian Responden Pasien.....	63
Tabel 6.10 Hasil Pengujian Responden Konselor	64
Tabel 6.11 Hasil Pengujian SUPER-Qm Pasien	64
Tabel 6.12 Hasil Pengujian SUPER-Qm Konselor.....	65
Tabel 6.13 Kritik dan Saran - Pasien	67
Tabel 6.14 Kritik dan Saran - Konselor.....	67
Tabel 7.1 Hasil Pengujian Usabilitas Pasien	76
Tabel 7.2 Hasil Pengujian Usabilitas Konselor	78

DAFTAR GAMBAR

Gambar 2.1 <i>Lifecycle</i> Aplikasi iOS.....	5
Gambar 2.2 SDLC - <i>Iterative Waterfall</i>	6
Gambar 2.3 Arsitektur MVVM	8
Gambar 2.4 Daftar Pertanyaan SUPR-Qm	10
Gambar 2.5 <i>Usability Rating Scale</i>	10
Gambar 3.1 Alur Penelitian Menggunakan <i>Waterfall</i>	11
Gambar 4.1 Usecase Diagram	18
Gambar 5.1 Arsitektur Sistem	26
Gambar 5.2 Sequence Diagram - Login	27
Gambar 5.3 Sequence Diagram - Register	28
Gambar 5.4 Sequence Diagram - Melihat Lokasi Konselor Dalam Peta.....	29
Gambar 5.5 Sequence Diagram - Melihat Daftar Konselor	30
Gambar 5.6 Sequence Diagram - Melihat Profile Konselor	30
Gambar 5.7 Sequence Diagram - Mengirim Permintaan Konseling	31
Gambar 5.8 Sequence Diagram - Melakukan Konseling	32
Gambar 5.9 Sequence Diagram - Memberi Rating ke Konselor	32
Gambar 5.10 Sequence Diagram - Melihat Riwayat Konseling	33
Gambar 5.11 Sequence Diagram - Membuat Appointment.....	34
Gambar 5.12 Sequence Diagram - Melihat Daftar Appointment Dengan Konselor	35
Gambar 5.13 Sequence Diagram - Melihat Daftar Permintaan Konseling.....	35
Gambar 5.14 Sequence Diagram - Melihat Profil Pasien	36
Gambar 5.15 Sequence Diagram - Merespon Permintaan Konseling.....	37
Gambar 5.16 Sequence Diagram - Melihat Daftar Appointment Dari Pasien	38
Gambar 5.17 Class Diagram Aplikasi Pasien	39
Gambar 5.18 Class Diagram - Aplikasi Sisi Konselor.....	40
Gambar 5.19 Screenflow Aplikasi Pasien.....	47
Gambar 5.20 Screenflow Aplikasi Konselor	48
Gambar 5.21 Implementasi Struktur Data.....	51
Gambar 5.22 Implementasi Antarmuka Pasien	55
Gambar 5.23 Implementasi Antarmuka Konselor.....	56
Gambar 6.1 <i>Usability Rating Scale</i>	68

DAFTAR LAMPIRAN

LAMPIRAN A HASIL EXPERT INTERVIEW	74
LAMPIRAN B HASIL PENGUJIAN USABILITAS	76

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kesehatan mental saat ini menjadi permasalahan yang tidak bisa diremehkan. Bahkan kesehatan mental sekarang menjadi masalah yang dihadapi banyak negara di dunia, tak terkecuali Indonesia. Data dari WHO mengatakan bahwa 804.000 jumlah kematian pada tahun 2012 disebabkan oleh kasus bunuh diri, dimana rasio bunuh diri yang terjadi di Indonesia mencapai 4,3 per 100.000 penduduk (WHO, 2012). Penderita kesehatan mental paling banyak dialami oleh kalangan remaja atau mahasiswa dan memiliki dampak negatif yang serius terhadap prestasi belajar, keluarga, dan lingkungan sekitar (Chen & Jiang, 2019).

Depresi menjadi penyebab utama dalam permasalahan kesehatan mental. Depresi memengaruhi 4,4% populasi dunia dan 5% populasi di Indonesia. Selain jumlah penderita depresi yang tinggi, masalah lain pada Indonesia adalah kurangnya jumlah ketersediaan tenaga profesional kesehatan mental seperti konselor dan psikiater. WHO pun menginisiasi sebuah program bernama *Mental Health Gap Action Program* (mhGAP) yang bertujuan untuk menyediakan *internet-based intervention* untuk kesehatan mental yang dapat didistribusikan secara luas. (Arjadi, Nauta, & Bockting, 2018).

Berbagai upaya telah dilakukan oleh pemerintah Indonesia dalam menangani permasalahan kesehatan mental, salah satunya adalah menerbitkan UU no 18, tentang kesehatan mental dan perawatan orang dengan penyakit mental yang dicakup oleh cakupan kesehatan secara *universal*. Namun, banyak orang masih mengalami kesulitan dalam mengakses layanan kesehatan mental (Tristiana et al., 2018). Di sisi lain, terdapat penelitian yang menunjukkan bahwa orang yang merasa depresi, sedih atau kesepian cenderung melampiaskan permasalahan tersebut dengan menggunakan media sosial, salah satunya adalah *Twitter* untuk sekedar bercerita atau mendapat dukungan interaksi (Mahoney et al., 2019). Hal ini menunjukkan bahwa *internet* dan teknologi informasi mampu menghasilkan sebuah solusi yang menjanjikan untuk melakukan perawatan masalah kesehatan mental yang ada di Indonesia (Arjadi et al., 2016). Mayoritas orang Indonesia menunjukkan keterbukaan untuk menggunakan *internet-based intervention* dalam menangani permasalahan kesehatan mental, dimana sangat penting untuk dilakukan sosialisasi pemanfaatan *internet* kepada masyarakat (Arjadi et al., 2018).

Dengan perkembangan teknologi yang semakin canggih dan digital, mampu menciptakan peluang untuk menyelesaikan permasalahan tersebut menggunakan sistem berbasis *mobile*. Dari data yang diperoleh, jumlah masyarakat di Indonesia yang menggunakan *smartphone* mencapai 41 juta pengguna dengan sistem operasi *Android*, sedangkan untuk sistem operasi *iOS* mencapai 2,8 juta (Rahman, 2015). Dengan memanfaatkan fitur lokasi pengguna yang ada pada *smartphone*, maka memberi kemudahan pada pengguna untuk mengetahui lokasi pengguna lain dan dapat saling bertukar informasi. Maka dari

itu, pemanfaatan aplikasi berbasis *mobile* dapat dilakukan untuk menghubungkan orang yang memiliki permasalahan dengan konselor terdekat untuk mendapatkan pertolongan pertama dalam masalahnya.

1.2 Rumusan Masalah

Dari latar belakang yang sudah dijelaskan diatas, maka rumusan masalah pada penelitian ini adalah sebagai berikut:

1. Apa saja kebutuhan fungsionalitas dan non-fungsional dari aplikasi yang dikembangkan?
2. Bagaimana hasil perancangan dari aplikasi yang dikembangkan?
3. Bagaimana hasil implementasi dari aplikasi yang dikembangkan?
4. Bagaimana hasil uji validasi dan usabilitas dari aplikasi yang dikembangkan?

1.3 Tujuan

Tujuan penelitian ini dilakukan adalah sebagai berikut:

1. Mengetahui apa saja kebutuhan fungsional dan non-fungsional pada aplikasi yang dikembangkan?
2. Mengetahui bagaimana hasil perancangan dari aplikasi yang dikembangkan?
3. Mengetahui bagaimana hasil implementasi dari aplikasi yang dikembangkan?
4. Mengetahui bagaimana hasil uji validasi dan usabilitas dari aplikasi yang dikembangkan?

1.4 Manfaat

1. Manfaat untuk penulis.

Mampu mengimplementasikan materi yang dipelajari pada perkuliahan untuk membangun aplikasi berbasis *mobile*.

2. Manfaat untuk peneliti selanjutnya.

Dapat menjadikan penelitian ini sebagai referensi atau acuan untuk membangun aplikasi berbasis *mobile*.

3. Manfaat untuk pengguna aplikasi

Dapat memberikan pertolongan pertama kepada penderita kesehatan mental dan memberikan pencegahan agar penderita segera membaik.

1.5 Batasan Masalah

Penelitian ini memiliki sejumlah batasan masalah sebagai berikut :

1. Aplikasi yang dibangun hanya menggunakan bahasa Indonesia.
2. Aplikasi harus terhubung ke internet agar bisa bekerja.
3. Aplikasi dibangun pada platform iOS dengan spesifikasi OS minimal iOS 13.

1.6 Sistematika Pembahasan

Struktur yang disusun pada proposal penelitian ini adalah sebagai berikut:

1. BAB I - Pendahuluan

Pada bab ini berisikan latar belakang masalah yang diangkat, perumusan masalah, tujuan dari penelitian yang dilakukan, batasan-batasan yang ada pada penelitian, dan sistematika pembahasan penelitian.

2. BAB II - Landasan Kepustakaan

Pada bab ini berisikan kajian-kajian kepustakaan yang digunakan sebagai referensi untuk penelitian.

3. BAB III - Metodologi

Pada bab ini berisikan penjelasan alur kerja penelitian yang dilakukan serta metodologi yang digunakan untuk menyelesaikan permasalahan.

4. BAB IV - Analisis Kebutuhan

Pada bab ini menjelaskan tentang proses menggali semua kebutuhan dari sistem yang akan dibangun.

5. BAB V - Perancangan dan Implementasi

Pada bab ini menjelaskan tentang proses perancangan untuk memulai pengembangan hingga hasil implementasi dari sistem yang dibangun.

6. BAB VI - Pengujian

Pada bab ini menjelaskan tentang proses pengujian dari sistem yang telah selesai diimplementasikan dan memperoleh hasil kesimpulan dari sistem yang telah dibuat.

7. BAB VII - Penutup

Pada bab ini memuat kesimpulan dari penelitian yang dilakukan dan berisi saran untuk peluang pengembangan lanjut kedepannya.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Dalam penelitian ini, penulis menemukan penelitian-penelitian serupa dan relevan yang sudah pernah dilakukan sehingga menjadi acuan dan referensi untuk penulis dalam melaksanakan penelitian.

Aplikasi pencarian rute perguruan tinggi berbasis android menggunakan *Location-Based Service*. Pada penelitian ini, penulis menawarkan solusi untuk memudahkan pengguna dalam mendapatkan informasi dan menentukan rute karena data dan informasi perguruan tinggi telah didaftarkan oleh peneliti (Nugroho, Nurhayati, & Widiyanto, 2017).

Aplikasi *mobile* berbasis lokasi untuk mencari layanan penyedia kesehatan di Yogyakarta. Pada penelitian ini, penulis membuat sebuah sistem yang mampu mencari lokasi layanan kesehatan dengan menggunakan *Google Map API* untuk menentukan rute dan lokasi terdekat (Erna Kumalasari Nurnawati, 2014).

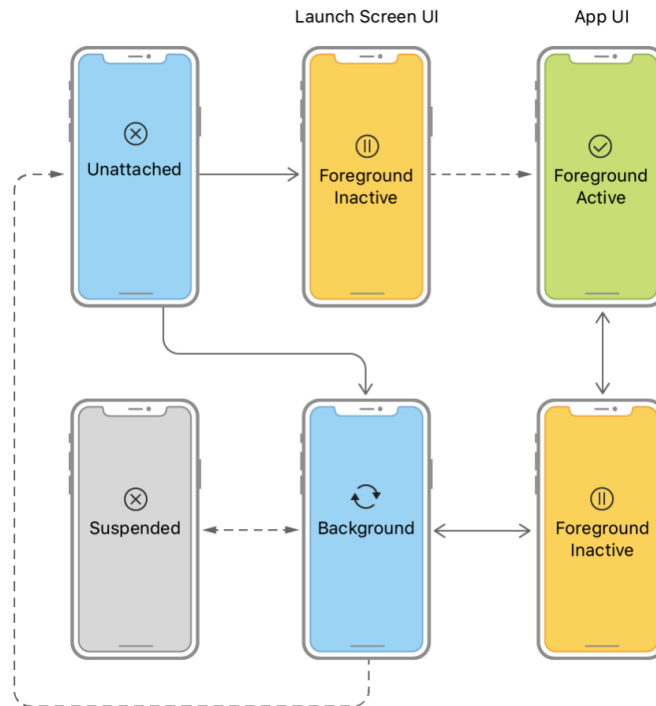
Pada negara Indonesia, *internet-based intervention* memiliki prospek yang menjanjikan dalam mengurangi kesenjangan kesehatan mental mengingat penggunaan internet dan smartphone terus meningkat tiap harinya. Untuk mengetahui apakah *internet-based intervention* diterima dan digunakan banyak orang, (Arjadi et al., 2018) melakukan investigasi untuk mengetahui faktor apa saja yang memengaruhi penggunaan *internet-based intervention* untuk kesehatan mental di Indonesia. Hasil dari penelitian ini adalah mayoritas masyarakat Indonesia memiliki keterbukaan dalam penggunaan *internet-based intervention* dalam menangani permasalahan kesehatan mental di Indonesia. Untuk meningkatkan adopsi intervensi berbasis internet, penting untuk terlebih dahulu mempromosikan penggunaan internet kepada lebih banyak orang di seluruh negeri, terutama bagi mereka yang saat ini sedang mengalami gangguan kesehatan mental.

2.2 Mobile Application

Mobile application adalah perangkat lunak atau sekumpulan program yang berjalan pada *mobile device* dan menjalankan tugas tertentu untuk penggunanya. *Mobile application* memiliki keunggulan diantaranya adalah kemudahan penggunaan, *user-friendly*, murah, mudah diunduh, dan berjalan hampir pada semua level perangkat bergerak. Selain itu, *mobile application* dapat digunakan untuk kebutuhan yang luas seperti menelpon, mengirim pesan, suara, video, game, dan lain-lain (Islam & Mazumder, 2010). Namun disamping kelebihan yang dimiliki, *mobile application* memiliki beberapa kelemahan diantaranya memiliki tampilan yang relatif kecil, memori yang terbatas, kemampuan komputasi pada CPU yang terbatas, rentang transfer data yang terbatas, dan lain-lain. (Oinas-Kukkonen & Kurkela, 2003)

2.2.1 iOS

iOS merupakan sistem operasi para perangkat *mobile* yang dibuat oleh Apple. Pada penelitian ini, sistem yang akan dibangun menggunakan bahasa *Swift* dan dibangun pada sistem operasi *iOS*. Diagram *lifeclyle* aplikasi dengan sistem operasi *iOS* digambarkan pada Gambar 2.1



Gambar 2.1 Lifecycle Aplikasi iOS

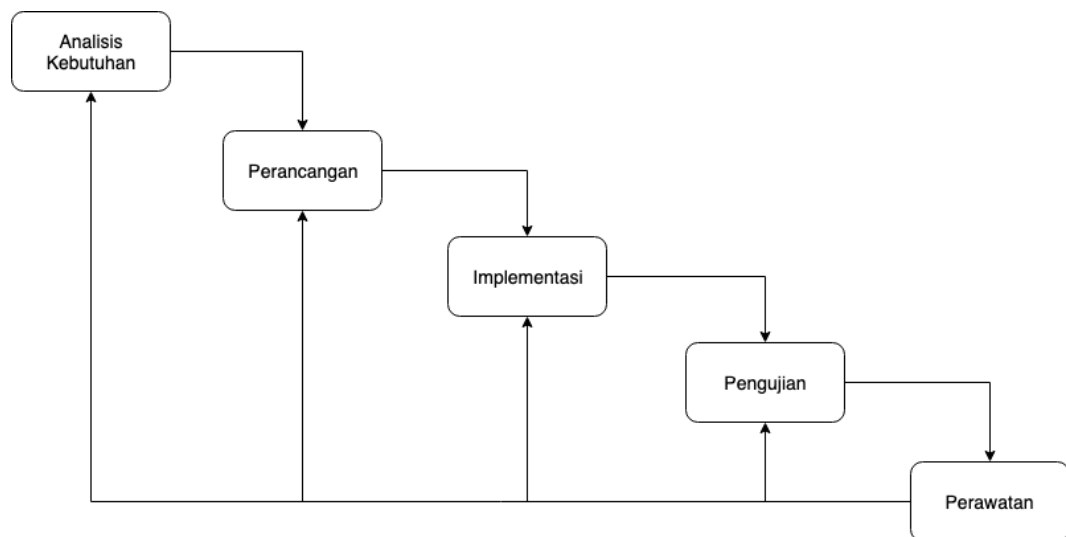
Sumber: Apple (2018)

Adapun kelebihan pengembangan aplikasi pada sistem operasi iOS menurut (Avram, 2013) adalah sebagai berikut :

1. Apple sudah menyediakan standar dalam pengembangan *user-interface* sehingga pengembang perangkat lunak dapat menghemat waktu dalam melakukan pengembangan aplikasi.
2. Sistem operasi iOS hanya berjalan pada *iPhone* dan *iPad* sehingga pengembang dapat membuat aplikasi yang *responsive* dengan lebih mudah.
3. Keamanan data dan privasi pada aplikasi berbasis iOS dapat terjaga dengan baik.
4. Biaya pengembangan aplikasi berbasis iOS relatif lebih murah dan mudah.

2.3 Iterative Waterfall Model

Waterfall merupakan siklus pengembangan perangkat lunak yang bersifat linear. SDLC (*Software Development Lifecycle*) *Waterfall* dibagi menjadi dua diantaranya *Classic Waterfall* dan *Iterative Waterfall*. Kelemahan dari *Classic Waterfall* adalah sulit untuk mengatur perubahan dan perawatan pada sistem. *Iterative Waterfall* merupakan pengembangan lanjut dari *Classic Waterfall* untuk memudahkan tim dan *developer* agar mampu menghasilkan sistem dengan waktu yang efisien dan memudahkan dalam merawat sistem (Rather & Bhatnagar, 2016). Fase pengembangan pada *Iterative Waterfall* digambarkan pada Gambar 2.2.



Gambar 2.2 SDLC - Iterative Waterfall

Sumber: Rather & Bhatnagar (2016)

Adapun tahap-tahap siklus pengembangan perangkat lunak menggunakan *Iterative Waterfall* adalah sebagai berikut:

1. Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap mendefinisikan kebutuhan fungsional dan kebutuhan non-fungsional akan sistem yang akan dikembangkan dengan melakukan wawancara dengan calon pengguna sistem.

2. Perancangan

Tahap perancangan merupakan tahap merancang segala sesuatu yang dibutuhkan dalam membangun sistem seperti algoritma, antarmuka, dan diagram-diagram untuk membantu memudahkan tahap implementasi.

3. Implementasi

Tahap implementasi merupakan tahap melakukan implementasi dari hasil perancangan yang telah selesai dibuat.

4. Pengujian

Tahap pengujian merupakan tahap melakukan pengujian dari sistem yang telah selesai untuk mengidentifikasi adanya *bug* sebelum sistem digunakan oleh pengguna.

5. Perawatan

Tahap perawatan adalah tahap ketika sistem telah rilis dan digunakan oleh pengguna, namun tim pengembang tetap melakukan perawatan apabila terdapat *bug* yang lolos dari tahap pengujian.

2.4 Location-Based Service

Location-based services atau layanan berbasis lokasi adalah layanan yang memperhitungkan lokasi geografis dari suatu objek (Junglas & Watson, 2008). Dengan adanya fitur *Global Positioning System* (GPS) pada perangkat *mobile* dapat membantu pengguna dalam melihat informasi lokasi suatu objek. Dalam konteks penelitian ini, peneliti menggunakan layanan berbasis lokasi dalam aplikasi untuk menentukan dan mencari konselor psikologi terdekat dengan pengguna.

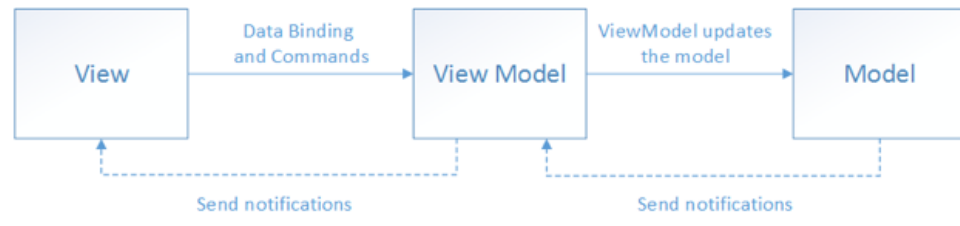
2.5 Best Practices

2.5.1 Swift

Swift adalah bahasa pemrograman baru yang dibuat oleh Apple pada tahun 2010 dan diumumkan pada tahun 2015 yang tujuannya utamanya digunakan untuk mengembangkan program di atas platform Apple (*iOS*, *macOS*, *tvOS* dan *watchOS*). *Swift* menawarkan kelebihan baru seperti *Type Safety*, *Fast* dan *Expressive*. *Type safety* yang adalah cara pemrograman baru agar program berjalan dengan cara yang aman (menghindari *null pointer*). *Swift* digunakan untuk menggantikan bahasa pemrograman yang dipakai Apple sebelumnya, yaitu *Objective-C*, yang secara performa lebih cepat dari bahasa pendahulunya. (Apple, n.d.-b)

2.5.2 MVVM (Model-View-ViewModel)

MVVM merupakan arsitektur perangkat lunak yang memiliki tiga *layer* yaitu *model*, *view*, dan *view model*. Dengan adanya tiga *layer* dalam sistem maka terdapat pemisahan antara *business-logic*, *presentation-logic*, dan *UI-logic*. Arsitektur ini awalnya dikembangkan oleh Microsoft untuk mengembangkan aplikasi pada platform mereka, kemudian banyak diadopsi oleh pengembang pada platform lainnya. Adapun alur interaksi antar *layer* digambarkan pada Gambar 2.3.



Gambar 2.3 Arsitektur MVVM

Sumber: Microsoft (2017)

2.5.3 MapKit

MapKit merupakan *framework* yang dibuat oleh Apple untuk menampilkan peta. Pada versi iOS 5.1 atau yang terbaru, *MapKit* menggunakan *Google Mobile Maps* (GMM) untuk menyediakan data yang sudah terintegrasi (Apple, n.d.-a). Pada penelitian ini, penulis menggunakan *MapKit* untuk menyediakan informasi lokasi konselor psikologi pada pengguna untuk memudahkan interaksi keduanya.

2.5.4 Firebase

Firebase merupakan penyedia layanan berbasis *cloud* yang dibuat oleh Google. Dengan adanya *Firebase*, pengembang perangkat lunak tidak perlu memikirkan untuk membuat *web-service* atau *API* untuk melakukan pengolahan data. Layanan *Firebase* yang digunakan pada penelitian ini adalah *Realtime Database* dan *Firebase Authentication*.

Implementasi *Firebase* pada pengembangan aplikasi perangkat bergerak memiliki keuntungan. Menurut (Mevada, 2018) keuntungan menggunakan *firebase* adalah sebagai berikut:

1. Penyimpanan data secara *real-time* dan sinkronisasi data
2. Adanya *Google Analytics* yang dapat membantu pengembang untuk melacak perilaku dari pengguna.
3. *Firebase* menyediakan fitur *crash reporting* pada pengguna sehingga pengembang dapat memperbaiki *bug* tersebut dengan cepat.
4. *Firebase Authentication* yang mempermudah pengembang dalam membuat fitur autentikasi pada aplikasi.

2.6 Pengujian Perangkat Lunak

2.6.1 BlackBox Testing

BlackBox Testing adalah pengujian yang dilakukan untuk mengevaluasi fitur atau kebutuhan dari pengguna dan memastikan bahwa sistem berjalan sesuai semestinya (Larrea, 2017). *Testcase* yang dirancang pada *blackbox testing* bergantung pada spesifikasi perangkat lunak yang telah ditetapkan.

2.6.2 Usability Testing

Usability Testing adalah pengujian yang dilakukan untuk menilai kualitas dari sistem oleh calon pengguna utama. Tujuan dilakukan *usability testing* adalah menemukan permasalahan yang dialami calon pengguna agar bisa diperbaiki untuk meningkatkan kenyamanan pengguna dalam mengoperasikan sistem nantinya. Tingkat *usability* dapat diketahui dengan melakukan pengukuran tentang sejauh mana tingkat kemudahan dan tingkat kepuasan dari pengguna (Farouqi, Aknuranda, & Herlambang, 2018).

Jumlah responden untuk *usability testing* dipilih 5 orang. Alasan pemilihan jumlah responden tersebut adalah 5 orang responden merupakan angka yang optimal dan efektif dimana mengacu pada keuntungan dan usaha yang dilakukan dalam pengujian (Nielson & Landauer, 1993). Alasan lain dipilih 5 orang sebagai jumlah responden adalah 95% permasalahan yang ada pada sistem akan muncul pada 5 orang pertama ketika dilakukan *usability testing* (Cao, 2015).

2.6.3 Skala Likert

Skala *likert* digunakan untuk mengukur pendapat seseorang terhadap sebuah pernyataan, dalam hal penelitian ini digunakan untuk mengukur tingkat kepuasan pengguna terhadap penggunaan sistem. Skala *likert* menggunakan interval yang memiliki rentang dari nilai 1 sampai 5, dimana nilai 1 merepresentasikan tidak setuju hingga nilai 5 merepresentasikan sangat setuju.

2.7 Standardized User Experience Percentile Rank Questionnaire for Mobile Apps (SUPR-Qm)

SUPR-Qm adalah instrumen pengukuran yang dikembangkan untuk melakukan pengujian usabilitas dari aplikasi perangkat bergerak. Pengembangan SUPR-Qm berawal dari kebutuhan pengukuran usabilitas untuk aplikasi perangkat bergerak yang awalnya hanya tersedia untuk aplikasi berbasis *desktop* dan *website* dengan menggunakan SUPR-Q. SUPR-Q digunakan untuk menangkap konstruksi pengalaman pengguna yang luas dan komponen yang lebih spesifik dari semua sistem berbasis *website* maupun *desktop* (Sauro & Zarolia, 2017). Daftar pertanyaan dari SUPR-Qm digambarkan pada Gambar 2.4.

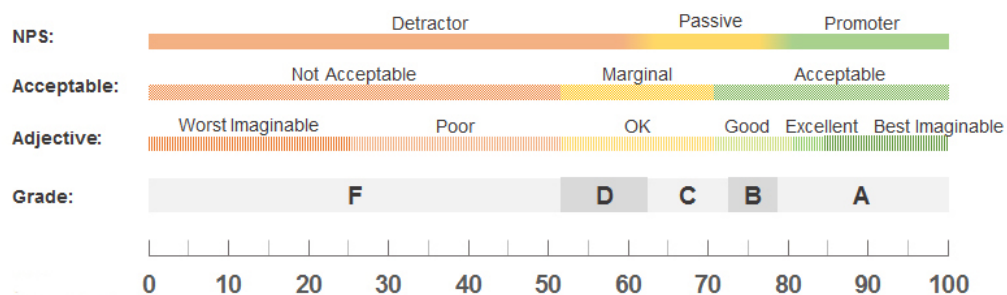
SUPR-Qm sendiri berbentuk daftar pertanyaan yang terdiri dari 16 buah dimana pertanyaan tersebut digunakan untuk mengukur tingkat usabilitas dari aplikasi perangkat bergerak dengan menggunakan skala *likert* sebagai komponen nilainya. Setelah pengujian selesai dilakukan maka dari hasil nilai SUPR-Qm dapat dikonversikan ke dalam tabel *usability rating scale* yang dijelaskan pada Gambar 2.5.

Table 1. Items Considered in Study 1

Abbreviation	Item	% N/A
Fun	It's fun using the app.	2%
Enjoy	I enjoy using the app.	2%
Happy	Using the app makes me happy.	2%
Exciting	It's exciting to use the app.	3%
Bored	I use the app when I am bored.	2%
LTR	How likely are you to recommend the app to a friend or colleague?	0%
MeetReq	The app's capabilities meet my requirements.	0%
MobileSite	The app offers features its mobile website doesn't.	22%
Crash	The app rarely crashes or causes problems on my phone.	0%
Bugs	The app runs without bugs or errors.	0%
Freq	I would like to use the app frequently.	0%
Misuse	The app does not misuse my information.	4%
Trust	I trust the app with my personal information.	2%
Appagain	I plan to use the app again soon.	1%
EasyNav	It is easy to navigate within the app.	1%
EasyUse	The app is easy to use.	0%
Attractive	I find the app to be attractive.	0%
Clean	The app has a clean and simple presentation.	1%
Discover	I like discovering new features on the app.	3%
Cantlive	I can't live without the app on my phone.	1%
Delightful	The app is delightful.	0%
LearnFriends	I talk about things I do or learn on the app with my friends.	3%
CxFriends	I am able to connect or communicate with friends directly from the app.	9%

Gambar 2.4 Daftar Pertanyaan SUPR-Qm

Sumber: Sauro & Zarolia (2017)

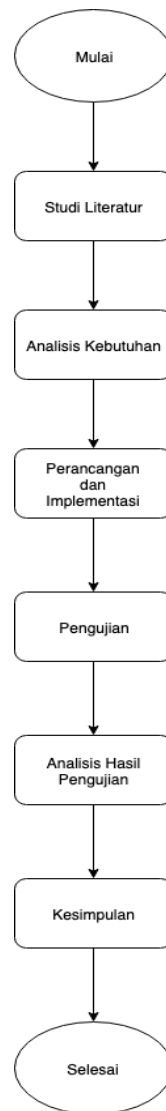


Gambar 2.5 Usability Rating Scale

Sumber: Sauro & Zarolia (2017)

BAB 3 METODOLOGI

Dalam sebuah penelitian diperlukan metodologi untuk menjadi landasan dan pedoman agar penelitian yang dilakukan terstruktur rapi. Metode yang digunakan pada penelitian yang dilakukan adalah *Waterfall*. Alur penelitian yang dilakukan digambarkan pada Gambar 3.1.



Gambar 3.1 Alur Penelitian Menggunakan *Waterfall*

3.1 Studi Literatur

Pada studi literatur, peneliti menemukan teori dan referensi penelitian yang relevan untuk memudahkan dalam melakukan pengembangan sistem. Adapun teori dan penelitian yang relevan sehingga menjadi referensi ketika penelitian ini dilakukan. Aplikasi pencarian rute perguruan tinggi berbasis android

menggunakan *Location-Based Service*. Pada penelitian ini, penulis menawarkan solusi untuk memudahkan pengguna dalam mendapatkan informasi dan menentukan rute karena data dan informasi perguruan tinggi telah didaftarkan oleh peneliti. (Nugroho et al., 2017).

Aplikasi *mobile* berbasis lokasi untuk mencari layanan penyedia kesehatan di Yogyakarta. Pada penelitian ini, penulis membuat sebuah sistem yang mampu mencari lokasi layanan kesehatan dengan menggunakan *Google Map API* untuk menentukan rute dan lokasi terdekat. (Erna Kumalasari Nurnawati, 2014)

Pada negara Indonesia, *internet-based intervention* memiliki prospek yang menjanjikan dalam mengurangi kesenjangan kesehatan mental mengingat penggunaan internet dan *smartphone* terus meningkat tiap harinya. Untuk mengetahui apakah *internet-based intervention* diterima dan digunakan banyak orang, (Arjadi et al., 2018) melakukan investigasi untuk mengetahui faktor apa saja yang memengaruhi penggunaan *internet-based intervention* untuk kesehatan mental di Indonesia. Hasil dari penelitian ini adalah mayoritas masyarakat Indonesia memiliki keterbukaan dalam penggunaan *internet-based intervention* dalam menangani permasalahan kesehatan mental di Indonesia. Untuk meningkatkan adopsi intervensi berbasis internet, penting untuk terlebih dahulu mempromosikan penggunaan internet kepada lebih banyak orang di seluruh negeri, terutama bagi mereka yang saat ini sedang mengalami gangguan kesehatan mental.

3.2 Analisis Kebutuhan

Pada tahap analisis kebutuhan, peneliti mendefinisikan semua kebutuhan fungsional dan non-fungsional yang didapatkan dari hasil wawancara untuk mengembangkan aplikasi menggunakan *iterative waterfall* sebagai metode pengembangan perangkat lunak.

3.3 Perancangan dan Implementasi

Pada tahap perancangan dan implementasi, dipastikan bahwa semua kebutuhan fungsional telah selesai dibuat sehingga implementasi dari tiap kebutuhan dapat segera dilakukan. Ketika tahap implementasi sudah selesai, diharapkan semua fitur telah selesai dibuat sehingga tahap pengembangan bisa dilanjutkan ke tahap pengujian dan *delivery* ke calon pengguna. Penjelasan masing-masing dari tahap perancangan dan implementasi adalah sebagai berikut:

3.3.1 Perancangan

Pada tahap perancangan, peneliti melakukan perancangan dasar terhadap sistem seperti membuat *class-diagram* untuk memodelkan arsitektur sistem dan *activity-diagram* yang digunakan untuk menggambarkan alur kerja sistem secara keseluruhan dari setiap *use-case* yang sudah ditetapkan. Kemudian dilanjutkan dengan membuat struktur data yang digunakan pada *Realtime Database* sebagai

layanan penyimpanan data yang digunakan, dan tipe autentikasi yang digunakan sistem pada *Firebase Authentication*.

Setelah tahap perancangan dasar sudah selesai, maka dilanjutkan dengan melakukan perancangan algoritma dari tiap *use-case* yang sudah ditetapkan. Hal ini dilakukan agar memudahkan peneliti dalam melanjutkan ke tahap implementasi nantinya.

3.3.2 Implementasi

Setelah tahap perancangan sudah selesai, maka dilakukan tahap implementasi ke pengembangan sistem. Hasil perancangan yang didapat menjadi pedoman untuk implementasi dilakukan mulai dari pembuatan *class* atau *interface* yang telah ditetapkan pada *class-diagram*, hingga integrasi sistem terhadap *Realtime Database* dan *Firebase Authentication*. Selanjutnya implementasi dari algoritme yang sudah ditetapkan dimana pada penelitian ini sistem dibangun pada perangkat bergerak dengan sistem operasi *iOS* menggunakan bahasa pemrograman *Swift*. Pada tahap akhir implementasi dilakukan perancangan *user-interface* menggunakan *framework* bawaan bernama *UIKit* yang telah disediakan oleh Apple.

3.4 Pengujian

Tahap pengujian pada penelitian ini adalah melakukan pengujian dari implementasi yang sudah selesai dilakukan. Pengujian dilakukan adalah *blackbox testing* agar memastikan fitur-fitur yang telah diimplementasi tidak memiliki *bug* atau masalah lainnya. Pada penelitian ini juga dilakukan pengujian *usability* yang dilakukan untuk mendapatkan umpan balik terhadap *user-experience* dari aplikasi yang dibangun. Dalam penelitian ini, jumlah responden yang ditetapkan untuk *usability testing* adalah 5 orang. Pengujian ini dilakukan dengan menggunakan SUPR-Qm sebagai instrumen yang terdiri dari daftar pertanyaan dan skala *likert* sebagai skala yang digunakan untuk penilaiannya.

3.4.1 Analisis Hasil Pengujian

Analisis hasil pengujian dilakukan setelah semua tahap pengujian sudah selesai. Analisis tersebut dilakukan untuk menilai hasil dari pengujian yang telah dilakukan. Analisis hasil pengujian meliputi hasil dari *blackbox testing* dan analisis hasil pengujian usability menggunakan Indeks Kepuasan dari hasil nilai menggunakan SUPR-Qm dan Skala *likert*.

3.5 Kesimpulan

Tahap kesimpulan merupakan tahap terakhir dimana semua kebutuhan fungsional dari sistem telah selesai diimplementasi dan pengujian telah selesai dilakukan. Hasil dari tahap analisis kebutuhan hingga pengujian merupakan kesimpulan dari penelitian yang dilakukan. Setelah kesimpulan didapatkan maka tahap terakhir adalah mendefinisikan saran evaluasi untuk penelitian berikutnya.

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini memuat proses analisis kebutuhan terhadap calon pengguna aplikasi yang berguna untuk mendapatkan kebutuhan sistem sehingga proses pengembangan dapat dimulai. Kebutuhan sistem yang dibuat mengacu pada hasil wawancara yang dilakukan terhadap calon pengguna dan konselor.

4.1 Gambaran Umum Sistem

Penelitian yang dilakukan adalah mengembangkan aplikasi perangkat bergerak untuk mencari konselor psikologi terdekat dengan pengguna dan dibangun pada *platform iOS*. Gambaran umum sistem terdiri dari deskripsi umum sistem dan lingkungan sistem.

4.1.1 Deskripsi Umum Sistem

Deskripsi umum pada sistem yang dibangun adalah aplikasi perangkat bergerak yang dikembangkan pada *platform iOS*. Aplikasi yang dikembangkan bertujuan untuk memudahkan pengguna dalam mencari konselor psikologi terdekat dengan pengguna sehingga pengguna dapat melakukan konseling secara *online* dan mempermudah pengguna untuk bertemu dengan konselor secara langsung. Setelah selesai melakukan konseling, pengguna dapat memberi *rating*, kritik dan saran kepada konselor.

Selain dari sisi pengguna, aplikasi dapat digunakan oleh konselor psikologi yang bertujuan untuk memudahkan konselor psikologi dalam melakukan konseling dengan pengguna. Konselor psikologi dapat melihat siapa saja pengguna yang ingin melakukan konseling. Kemudian konselor psikologi memiliki pilihan untuk menerima atau menolak permintaan dari pengguna untuk konselor. Konselor psikologi juga dapat melihat profil dari pengguna sebelum menerima permintaan konseling dari pengguna.

4.1.2 Lingkungan Sistem

Aplikasi ini dikembangkan menggunakan bahasa pemrograman *swift* dan *Xcode* sebagai IDE (*Integrated Development Environment*). Peneliti memanfaatkan *MapKit*, sebuah *framework* buatan *Apple* untuk dapat menampilkan peta pada aplikasi dan menggunakan *CLLocation* untuk mendapatkan informasi lokasi pengguna dan juga konselor psikologi. Untuk pengolahan data dan menangani *request* dari aplikasi, peneliti menggunakan *Firebase* dimana layanan yang digunakan adalah *firebase authentication* dan *realtime database*.

4.2 Analisis Kebutuhan Perangkat Lunak

4.2.1 Identifikasi Aktor

Definisi aktor adalah semua orang atau sistem eksternal yang memiliki peran secara langsung terhadap sistem. Aktor pada aplikasi ini terdiri dari *guest*, *user*, dan konselor. Penjelasan aktor dan tiap deskripsinya dapat dilihat pada Tabel 4.1.

Tabel 4.1 Daftar Aktor

Aktor	Deskripsi
<i>Guest</i>	Pada dasarnya, <i>guest</i> adalah pengguna namun belum memiliki akun atau belum melakukan autentikasi. Aktor <i>guest</i> dapat melihat aplikasi namun memerlukan proses autentikasi untuk mengakses fitur-fitur tertentu.
Pasien	Pasien adalah pengguna yang telah teridentifikasi melalui proses autentikasi pada aplikasi. Aktor pasien memiliki wewenang untuk mengakses semua fitur dari aplikasi yang dikembangkan.
Konselor	Aktor konselor adalah aktor yang telah teridentifikasi sebagai konselor setelah melakukan proses autentikasi.

4.2.2 Kebutuhan Fungsional Sistem

Kebutuhan fungsional sistem adalah daftar yang mendeskripsikan ekspektasi layanan yang seharusnya dilakukan oleh sistem (Faisandier, 2012). Pada penelitian ini, kebutuhan fungsional sistem dibuat berdasarkan hasil wawancara yang telah didapatkan. Dalam membuat daftar kebutuhan fungsional, tiap kebutuhan diberi kode sebagai identitas untuk memudahkan peneliti dalam mereferensikannya dalam seluruh proses penelitian. Daftar kebutuhan fungsional sistem untuk pengguna dapat dilihat pada Tabel 4.2 dan kebutuhan fungsional sistem untuk konselor dapat dilihat pada Tabel 4.3.

Tabel 4.2 Kebutuhan Fungsional - Pasien

No	Kode Fungsi	Deskripsi Kebutuhan
1.	SS-F-001	Sistem dapat menyediakan tempat untuk pasien melakukan <i>login</i> dengan mengisi email dan kata sandi pada tempat yang disediakan.
2.	SS-F-002	Sistem dapat menyediakan tempat untuk pasien melakukan <i>register</i> dengan mengisi nama, tanggal lahir, jenis kelamin, email dan kata sandi pada tempat yang disediakan.
3.	SS-F-003	Sistem dapat menampilkan peta yang berisi lokasi pasien dan lokasi konselor psikologi terdekat dengan

		pasien.
4.	SS-F-004	Sistem dapat menampilkan daftar semua konselor psikologi terdekat dengan pasien.
5.	SS-F-005	Sistem dapat menampilkan detail profil konselor psikologi yang dipilih pasien.
6.	SS-F-006	Sistem dapat memfasilitasi pasien untuk mengirim permintaan konseling dengan konselor psikologi yang dipilih pasien.
7.	SS-F-007	Sistem dapat menampilkan halaman <i>chatting</i> antara pasien dan konselor dalam melakukan konseling secara <i>online</i> .
8.	SS-F-008	Sistem dapat memfasilitasi pasien untuk memberi rating kepuasan terhadap konselor setelah selesai melakukan konseling.
9.	SS-F-009	Sistem dapat menampilkan halaman riwayat konseling antara pasien dan konselor psikologi.
10.	SS-F-010	Sistem dapat memfasilitasi pasien untuk membuat <i>appointment</i> dengan konselor psikologi.

Tabel 4.3 Kebutuhan Fungsional - Konselor

1.	SS-KF-001	Sistem dapat menampilkan seluruh permintaan konseling dari pasien untuk konselor psikologi.
2.	SS-KF-002	Sistem dapat memfasilitasi konselor untuk menerima atau menolak permintaan konseling dari pasien.
3.	SS-KF-003	Sistem dapat menampilkan profil pasien yang mengirim permintaan konseling untuk konselor psikologi.
4.	SS-KF-004	Sistem dapat menampilkan daftar <i>appointment</i> dari pasien.

4.2.3 Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional adalah kebutuhan yang menetapkan atribut pada sistem perangkat lunak seperti batasan lingkungan dan implementasi pada sistem, performa sistem, keandalan sistem, dan perawatan pada sistem (Bajpai & Gorthi, 2012). Kebutuhan non-fungsional yang dibuat pada aplikasi yang dikembangkan pada penelitian ini mencakup *usability* yang diukur dengan melakukan pengujian kepada pengguna untuk melakukan uji skenario terhadap aplikasi yang sudah ditetapkan. Adapun ekspektasi minimal dari nilai kepuasan pada aplikasi yang dikembangkan adalah cukup memuaskan atau *ok*.

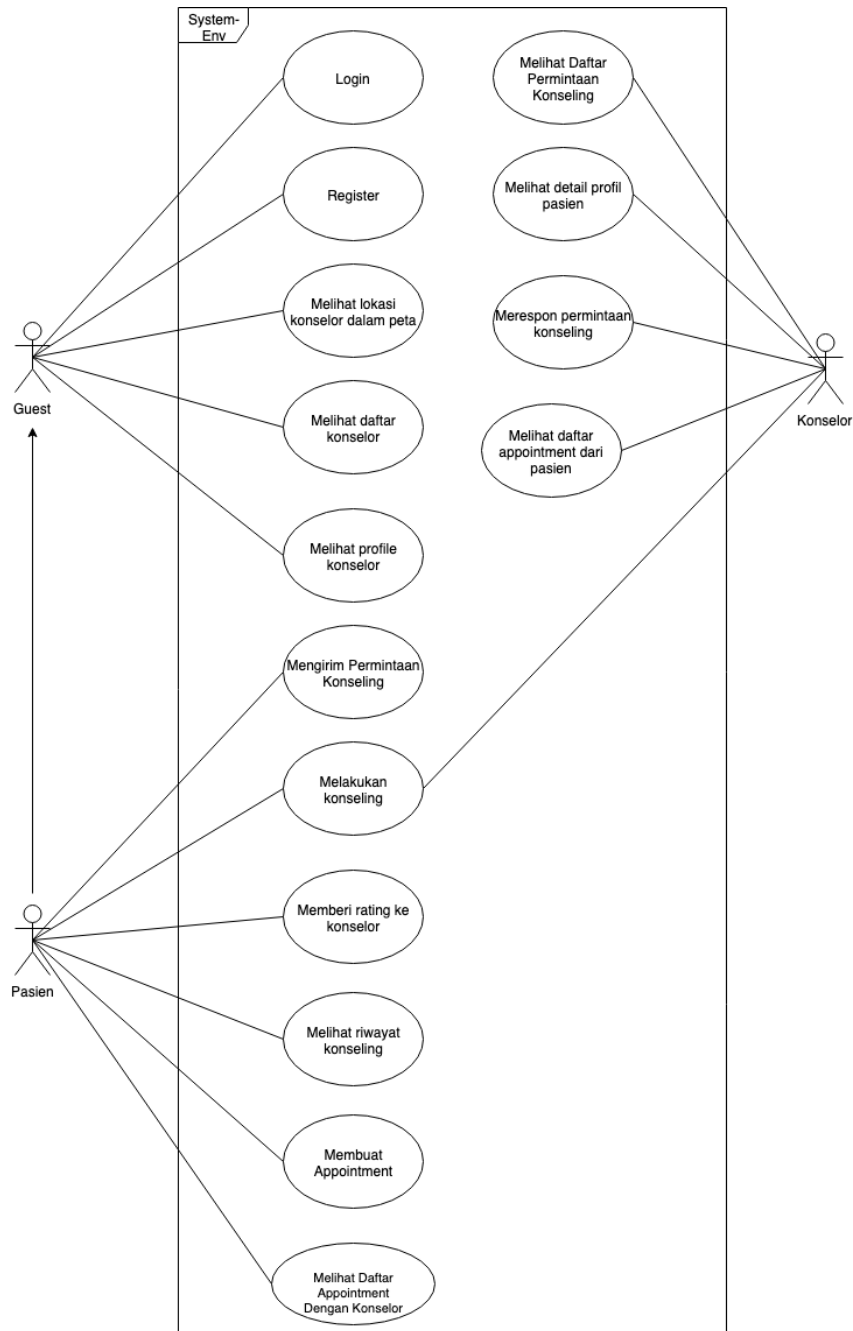
4.2.4 Usecase Diagram

Usecase diagram adalah diagram yang menggambarkan interaksi secara abstrak antara aktor dengan sistem (Shivaram & Handigund, 2015). Pada penelitian ini terdapat tiga aktor dengan masing-masing peran yang berbeda. *Usecase* diagram pada penelitian ini dapat dilihat pada Gambar 4.1. Pada *usecase* diagram yang telah dibuat, terdapat tiga aktor yang terlibat didalam sistem yaitu *guest*, pasien, dan konselor.

Aktor *guest* dapat melakukan *login* dengan mengisi *email* dan *password* pada tempat yang telah disediakan oleh sistem. Kemudian pada *usecase register*, aktor *guest* dapat melakukannya dengan mengisi nama lengkap, tanggal lahir, jenis kelamin, *email*, dan *passwordnya*. Pada *usecase* melihat lokasi konselor dalam peta, aktor *guest* dapat melihat semua lokasi konselor yang terdekat dengannya. Pada *usecase* melihat daftar konselor, aktor *guest* dapat melihat semua konselor dalam sistem yang berbentuk *list* dan diurutkan berdasarkan jarak antara aktor dengan konselor. Pada *usecase* melihat profil konselor, aktor *guest* dapat memilih konselor kemudian sistem menampilkan detail informasi mengenai konselor tersebut.

Aktor pasien merupakan spesialisasi dari aktor *guest*, dimana aktor pasien dapat melakukan semua aktivitas atau *usecase* yang dimiliki pada aktor *guest* ditambah dengan *usecase* khusus pada aktor pasien itu sendiri. Pada *usecase* mengirim permintaan konseling, aktor pasien dapat mengirim permintaan untuk melakukan konseling secara online dengan konselor yang dipilihnya. Pada *usecase* melakukan konseling secara online, setelah aktor pasien mengirim permintaan konseling kepada konselor dan konselor tersebut menerima permintaannya maka aktor pasien dapat melakukan konseling secara *online* dengan konselor tersebut. Pada *usecase* memberi rating ke konselor, aktor pasien dapat memberikan rating kepuasan terhadap konselor setelah selesai melaksanakan konseling. Pada *usecase* melihat daftar riwayat konseling, aktor pasien dapat melihat semua riwayat konseling yang pernah dilakukan sebelumnya dengan konselor.

Aktor konselor memiliki tiga *usecase* pada aplikasi yang dikembangkan. Pada *usecase* melihat daftar permintaan konseling, aktor konselor dapat melihat semua permintaan konseling yang masuk kedalam akunnya. Pada *usecase* melihat detail profil pasien, aktor konselor dapat melihat detail informasi profil dari pasien yang mengirim permintaan untuk melakukan konseling. Pada *usecase* merespon permintaan konseling, aktor konselor dapat menerima permintaan konseling maupun menolak permintaan konseling dari pasien.



Gambar 4.1 Usecase Diagram

4.2.5 Usecase Scenario

Usecase scenario mendefinisikan skenario aktivitas aktor dalam melaksanakan *usecase*. Berikut merupakan *usecase scenario* yang dikembangkan pada penelitian yang dilakukan.

Tabel 4.4 Usecase Scenario - Login

<i>Usecase</i>	Login
<i>Actor</i>	<i>Guest</i>
<i>Target</i>	<i>Guest</i> telah teridentifikasi sebagai <i>user</i> dalam sistem
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. <i>Guest</i> mengisi email dan password pada field yang disediakan dan menekan tombol login. 2. Sistem melakukan validasi email dan password dari <i>guest</i>, kemudian menampilkan halaman utama.
<i>Alternative Flow</i>	Apabila sistem gagal melakukan validasi email dan password pengguna, maka sistem akan menampilkan pesan error.
<i>Post Condition</i>	<i>Guest</i> sudah teridentifikasi sebagai pasien dan masuk kedalam halaman utama sistem.

Tabel 4.5 Usecase Scenario - Register

<i>Usecase</i>	Register
<i>Actor</i>	<i>Guest</i>
<i>Target</i>	<i>Guest</i> memiliki akun yang terdaftar
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman login. 2. <i>Guest</i> menekan tombol navigasi ke halaman <i>register</i>. 3. Sistem menampilkan halaman register. 4. <i>Guest</i> mengisi data diri yang terdiri dari nama lengkap, tanggal lahir, jenis kelamin, email, dan password. Kemudian <i>guest</i> menekan tombol register. 5. Sistem memvalidasi data diri dari <i>guest</i>, kemudian menampilkan ke halaman utama.
<i>Alternative Flow</i>	Apabila sistem gagal memvalidasi data diri, maka sistem akan menampilkan pesan error.

<i>Post Condition</i>	<i>Guest</i> memiliki akun yang terdaftar dan masuk kedalam halaman utama sistem.
-----------------------	---

Tabel 4.6 Usecase Scenario - Melihat lokasi konselor dalam peta

<i>Usecase</i>	Melihat lokasi konselor dalam peta
<i>Actor</i>	<i>Guest</i>
<i>Target</i>	<i>Guest</i> dapat melihat semua lokasi konselor dalam peta.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. <i>Guest</i> membuka <i>tab</i> beranda pada sistem 2. Sistem menampilkan semua lokasi konselor dalam peta
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan semua lokasi konselor dalam peta.

Tabel 4.7 Usecase Scenario - Melihat daftar konselor

<i>Usecase</i>	Melihat daftar konselor
<i>Actor</i>	<i>Guest</i>
<i>Target</i>	<i>Guest</i> melihat daftar konselor terdekat.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. <i>Guest</i> menekan tombol tampilkan semua konselor 2. Sistem menampilkan daftar konselor terdekat dengan pengguna.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan daftar konselor terdekat.

Tabel 4.8 Usecase Scenario - Melihat profile konselor

<i>Usecase</i>	Melihat profil konselor
<i>Actor</i>	<i>Guest</i>
<i>Target</i>	<i>Guest</i> dapat melihat detail informasi profil konselor yang dipilih.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. <i>Guest</i> memilih konselor. 2. Sistem menampilkan detail informasi profil konselor
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan detail informasi profil dari konselor yang dipilih.

Tabel 4.9 Usecase Scenario - Mengirim permintaan konseling

<i>Usecase</i>	Mengirim permintaan konseling
<i>Actor</i>	Pasien
<i>Target</i>	Pasien dapat mengirim permintaan konseling kepada konselor yang dipilih..
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pasien menekan tombol kirim permintaan konseling pada konselor yang dipilih 2. Sistem menampilkan halaman menunggu konfirmasi konselor
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah mengirim permintaan konseling dari pasien kepada konselor yang telah dipilih.

Tabel 4.10 Usecase Scenario - Melakukan konseling

<i>Usecase</i>	Melakukan konseling
<i>Actor</i>	Pasien
<i>Target</i>	Pasien melakukan konseling secara online dengan <i>chatting</i> dengan konselor.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Sistem menampilkan halaman <i>chatting</i>.

	2. Pasien mengirim pesan. 3. Sistem menampilkan pesan dari pasien 4. Pasien menekan tombol selesai konseling.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Pasien telah melakukan konseling secara online dengan konselor.

Tabel 4.11 Usecase Scenario - Memberikan rating kepada konselor

<i>Usecase</i>	Memberi rating kepada konselor
<i>Actor</i>	Pasien
<i>Target</i>	Pasien memberi rating kepada konselor setelah selesai melakukan konseling.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	1. Sistem menampilkan halaman rating. 2. Pasien memberi rating dan menekan tombol selesai
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem kembali menampilkan halaman utama sistem

Tabel 4.12 Usecase Scenario - Membuat Appointment

<i>Usecase</i>	Membuat <i>Appointment</i>
<i>Actor</i>	Pasien
<i>Target</i>	Pasien dapat membuat <i>appointment</i> dengan konselor psikologi
<i>Pre-Condition</i>	-
<i>Main Flow</i>	1. Sistem menampilkan halaman <i>appointment</i> 2. Pasien membuat <i>appointment</i> dari waktu yang tersedia pada hari ini 3. Sistem menampilkan pesan sukses membuat <i>appointment</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah mengirim permintaan konseling dari

	pasien kepada konselor yang telah dipilih.
--	--

Tabel 4.13 Usecase Scenario - Melihat Riwayat Konseling

<i>Usecase</i>	Melihat riwayat konseling
<i>Actor</i>	Pasien
<i>Target</i>	Pasien dapat melihat semua riwayat konseling.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pasien melihat riwayat konseling 2. Sistem menampilkan daftar riwayat konseling dari pasien.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan daftar riwayat konseling dari pasien.

Tabel 4.14 Usecase Scenario - Melihat Daftar Appointment Dengan Konselor

<i>Usecase</i>	Melihat Daftar <i>Appointment</i> dengan Konselor
<i>Actor</i>	Pasien
<i>Target</i>	Pasien dapat melihat daftar <i>appointment</i> dengan konselor.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Pasien melihat daftar <i>appointment</i> 2. Sistem menampilkan daftar <i>appointment</i> dengan konselor
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan daftar <i>appointment</i> dengan konselor.

Tabel 4.15 Usecase Scenario - Melihat daftar permintaan konseling

<i>Usecase</i>	Melihat daftar permintaan konseling
<i>Actor</i>	Konselor
<i>Target</i>	Konselor dapat melihat semua permintaan konseling dari pasien.
<i>Pre-Condition</i>	-

<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konselor membuka <i>tab</i> konseling 2. Sistem menampilkan daftar permintaan konseling dari pasien.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan semua permintaan konseling dari pasien.

Tabel 4.16 Usecase Scenario - Melihat detail profil user

<i>Usecase</i>	Melihat detail profil pasien
<i>Actor</i>	Konselor
<i>Target</i>	Konselor melihat detail informasi dari profil pasien yang mengirim permintaan konseling.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konselor memilih permintaan konseling 2. Sistem menampilkan detail profil dari pasien.
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan detail informasi profile dari pasien.

Tabel 4.17 Usecase Scenario - Melihat Daftar *Appointment* Dari Pasien

<i>Usecase</i>	Melihat daftar <i>appointment</i> dari pasien
<i>Actor</i>	Konselor
<i>Target</i>	Konselor dapat melihat daftar <i>appointment</i> dari pasien.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konselor membuka halaman daftar <i>appointment</i> 2. Sistem menampilkan halaman <i>appointment</i>
<i>Alternative Flow</i>	-
<i>Post Condition</i>	Sistem telah menampilkan semua permintaan konseling dari pasien.

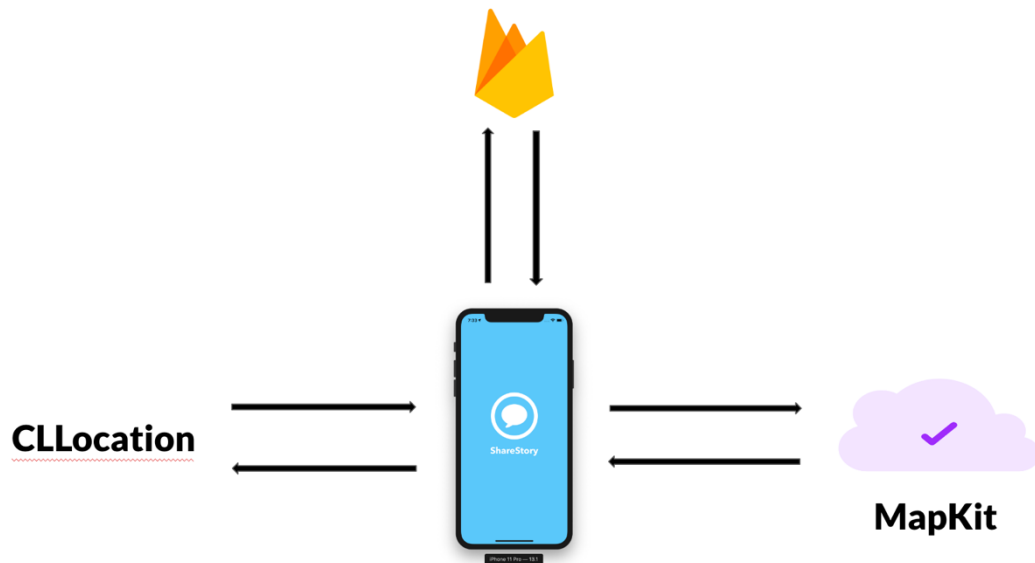
Tabel 4.18 Usecase Scenario - Merespon permintaan konseling

<i>Usecase</i>	Merespon permintaan konseling
<i>Actor</i>	Konselor
<i>Target</i>	Konselor memberi respon kepada permintaan konseling yang dikirim pasien.
<i>Pre-Condition</i>	-
<i>Main Flow</i>	<ol style="list-style-type: none"> 1. Konselor menekan tombol untuk merespon permintaan konseling dari pasien. 2. Sistem menampilkan halaman <i>chatting</i>.
<i>Alternative Flow</i>	Apabila konselor menolak permintaan dari pasien, maka sistem menampilkan pesan <i>pop-up</i> berisi alasan penolakan dari konselor.
<i>Post Condition</i>	Sistem telah mengirim respon dari konselor kepada pasien.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Bab ini membahas tentang perancangan yang dilakukan untuk membuat pemodelan dari sistem yang akan dibuat. Hasil dari perancangan akan diimplementasikan dan dibahas pada bab implementasi. Tahap perancangan terdiri dari perancangan arsitektur sistem, *sequence diagram*, *class diagram*, perancangan basis data, dan diakhiri dengan membuat rancangan antarmuka.

5.1 Arsitektur Sistem



Gambar 5.1 Arsitektur Sistem

Perancangan arsitektur sistem pada aplikasi yang dikembangkan digambarkan pada Gambar 5.1. Pada arsitektur sistem yang dibangun terdapat tiga komponen *library* utama yaitu *MapKit*, *CLLocation*, dan *Firebase*. *MapKit* adalah *framework* buatan Apple yang digunakan untuk menampilkan peta. Dalam hal ini, *MapKit* digunakan untuk menampilkan lokasi pasien dan konselor psikologi terdekat dengan pasien. *CLLocation* adalah *native-library* yang digunakan untuk mengatur dan memonitor lokasi terkini dari pasien dan mendapatkan alamat fisik dari konselor psikologi. *Firebase* adalah layanan berbasis *cloud* yang dibuat oleh Google. Pada penelitian ini, layanan *Firebase* yang digunakan adalah *Realtime Database* yang bertujuan untuk melakukan transaksi pertukaran data dengan sisi aplikasi serta memanfaatkan fitur *realtime* untuk membuat layanan *chatting* pada sisi aplikasi.

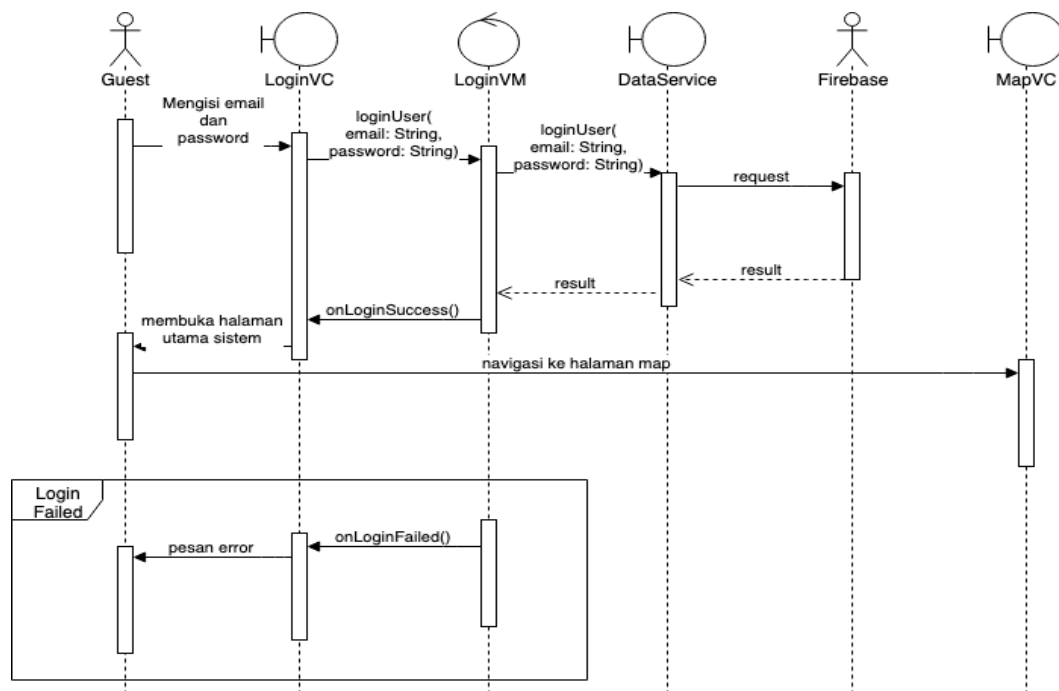
Selain menggunakan ketiga *library* tersebut, arsitektur yang digunakan untuk mengorganisir tiap modul dan fitur adalah MVVM (*Model-View-ViewModel*). *Model* memiliki tugas untuk membuat representasi data objek. *ViewModel* memiliki tugas untuk menyediakan data yang siap dikonsumsi untuk *view*. Dan *view* memiliki tugas untuk menampilkan data dan menerima respon atau *event* dari pengguna. Dalam penerapannya, terdapat penambahan satu layer yaitu

Service, dimana sebuah objek *singleton* yang memiliki tugas untuk melakukan *request* ke *Firebase* untuk melakukan pertukaran data.

5.2 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan urutan aktifitas antara aktor dengan komponen pada sistem pada masing-masing kebutuhan fungsional. Berikut merupakan hasil perancangan *sequence diagram* dari kebutuhan fungsional yang telah didefinisikan.

5.2.1 Sequence Diagram - Login

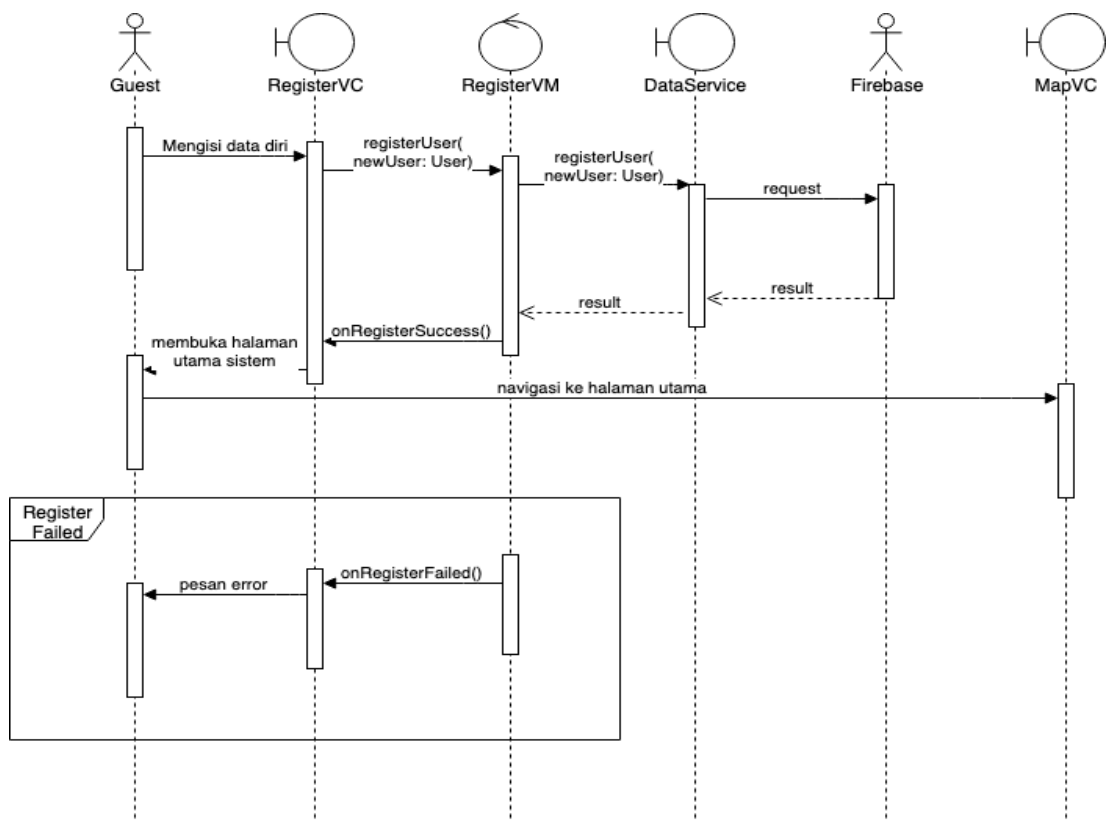


Gambar 5.2 Sequence Diagram - Login

Sequence diagram login dari aktor *Guest* digambarkan pada Gambar 5.2. Aktor *guest* mengisi email dan password pada tempat yang disediakan pada halaman *LoginVC*. Kemudian pada *class LoginVC* memanggil fungsi `loginUser(email: String, password: String)` yang ada pada *class LoginVM*. Dari *LoginVM* kemudian memanggil fungsi `loginUser(email: String, password: String)` yang ada pada *class DataService*. Pada *class DataService* melakukan *request* kepada *Firebase* dan mendapatkan *result* berupa hasil verifikasi login dari aktor *guest*. Apabila proses login berhasil maka *LoginVM* memanggil fungsi `onLoginSuccess()` pada *class LoginVC* dan mengarahkan aktor untuk berpindah ke halaman *MapVC*. Apabila proses login gagal maka sistem menampilkan pesan *error*.

5.2.2 Sequence Diagram - Register

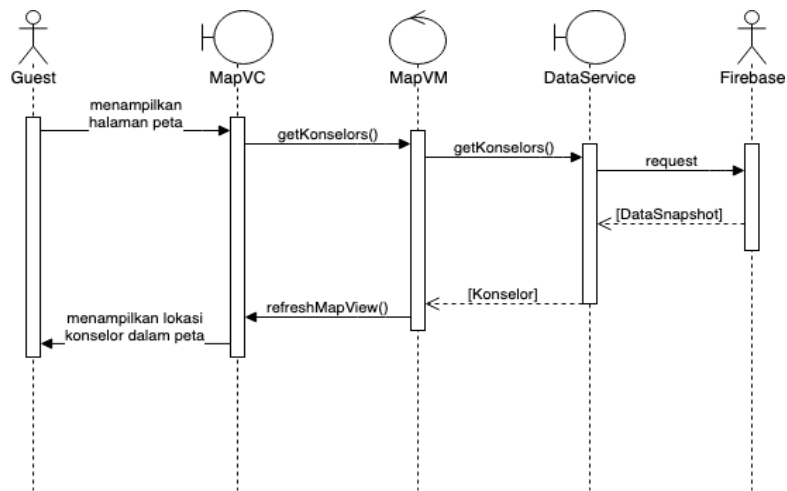
Sequence diagram register dari aktor *Guest* digambarkan pada Gambar 5.3. Aktor *guest* mengisi data diri pada tempat yang disediakan pada halaman *RegisterVC*. Kemudian pada class *RegisterVC* memanggil fungsi `registerUser(newUser: User)` yang ada pada class *registerVM*. Dari *registerVM* kemudian memanggil fungsi `registerUser(newUser: User)` yang ada pada class *DataService*. Pada class *DataService* melakukan request kepada *Firebase* dan mendapatkan *result* berupa hasil verifikasi register dari aktor *guest*. Apabila proses login berhasil maka *RegisterVM* memanggil fungsi `onRegisterSuccess()` pada class *RegisterVC* dan mengarahkan aktor untuk berpindah ke halaman *MapVC*. Apabila proses register gagal maka sistem menampilkan pesan *error*.



Gambar 5.3 Sequence Diagram - Register

5.2.3 Sequence Diagram - Melihat Lokasi Konselor Dalam Peta

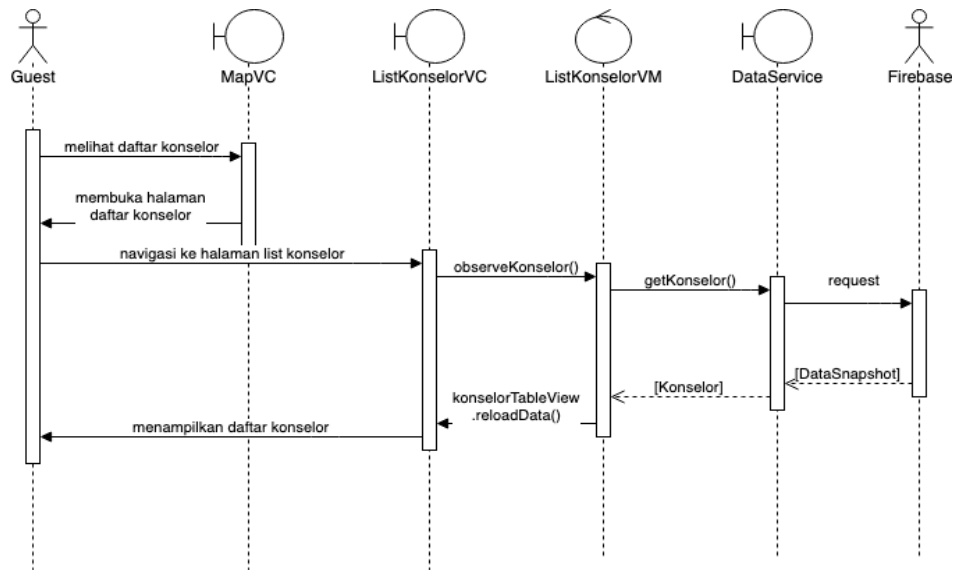
Sequence diagram melihat lokasi konselor dalam peta oleh aktor *guest* digambarkan pada Gambar 5.4. Sistem menampilkan halaman *MapVC* pada aktor yang kemudian memanggil fungsi `getKonselors()` pada objek *MapVM*. Dari *MapVM* kemudian memanggil fungsi `getKonselors()` yang ada pada objek *DataService*. Kemudian objek *DataService* melakukan *request* kepada *Firestore* untuk mendapatkan data konselor. Ketika *DataService* telah selesai melakukan *request* dan mengembalikan data menuju objek *MapVM* maka dilakukan pemanggilan fungsi `refreshMapView()` yang ada pada halaman *MapVC*.



Gambar 5.4 Sequence Diagram - Melihat Lokasi Konselor Dalam Peta

5.2.4 Sequence Diagram - Melihat Daftar Konselor

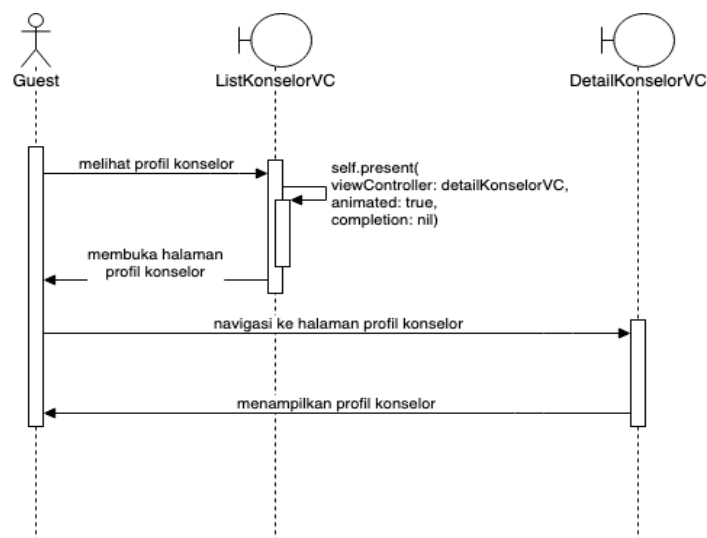
Sequence diagram melihat daftar konselor dari aktor *guest* digambarkan pada Gambar 5.5. Pada halaman *MapVC*, aktor *guest* menekan tombol yang tersedia untuk melihat daftar konselor dan berpindah halaman ke *ListKonselorVC*. Kemudian memanggil fungsi `observeKonselor()` yang ada pada objek *ListKonselorVM*. Dari objek *ListKonselorVM* kemudian memanggil fungsi `getKonselor()` yang ada pada objek *DataService*. Setelah itu objek *DataService* melakukan *request* ke *Firestore* dan menghasilkan data konselor yang dikembalikan ke objek *ListKonselorVM*. Data yang dihasilkan membuat halaman *ListKonselorVC* melakukan `reloadData()` untuk menampilkan data konselor kepada aktor.



Gambar 5.5 Sequence Diagram - Melihat Daftar Konselor

5.2.5 Sequence Diagram - Melihat Profile Konselor

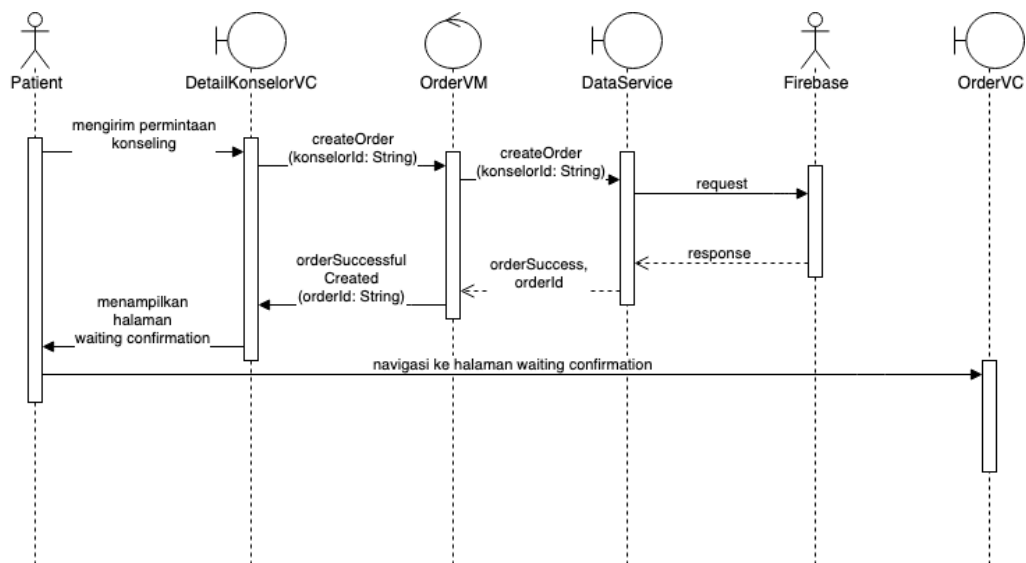
Sequence diagram melihat profile konselor oleh aktor *guest* digambarkan pada Gambar 5.6. Pada halaman *ListKonselorVC*, aktor *guest* memilih konselor yang tersedia untuk dapat melihat detail profilnya. Kemudian pada sistem membuka halaman profil konselor yang bernama *DetailKonselorVC*.



Gambar 5.6 Sequence Diagram - Melihat Profile Konselor

5.2.6 Sequence Diagram - Mengirim Permintaan Konseling

Sequence diagram mengirim permintaan konseling oleh aktor pasien digambarkan pada Gambar 5.7. Pasien menekan tombol untuk mengirim permintaan konseling pada halaman *DetailKonselorVC*. Kemudian memanggil fungsi `createOrder(konselorId: String)` pada objek *OrderVM*. *OrderVM* kemudian memanggil fungsi `createOrder(konselorId: String)` pada objek *DataService*. Setelah pemanggilan fungsi tersebut maka objek *DataService* melakukan *request* ke *Firestore* dan mendapatkan respon berupa `orderSuccess` dan `orderId`. Kemudian objek *OrderVM* memanggil fungsi `orderSuccessfulCreated(orderId: String)` pada *DetailKonselorVC* dan menampilkan halaman menunggu konfirmasi dari konselor.

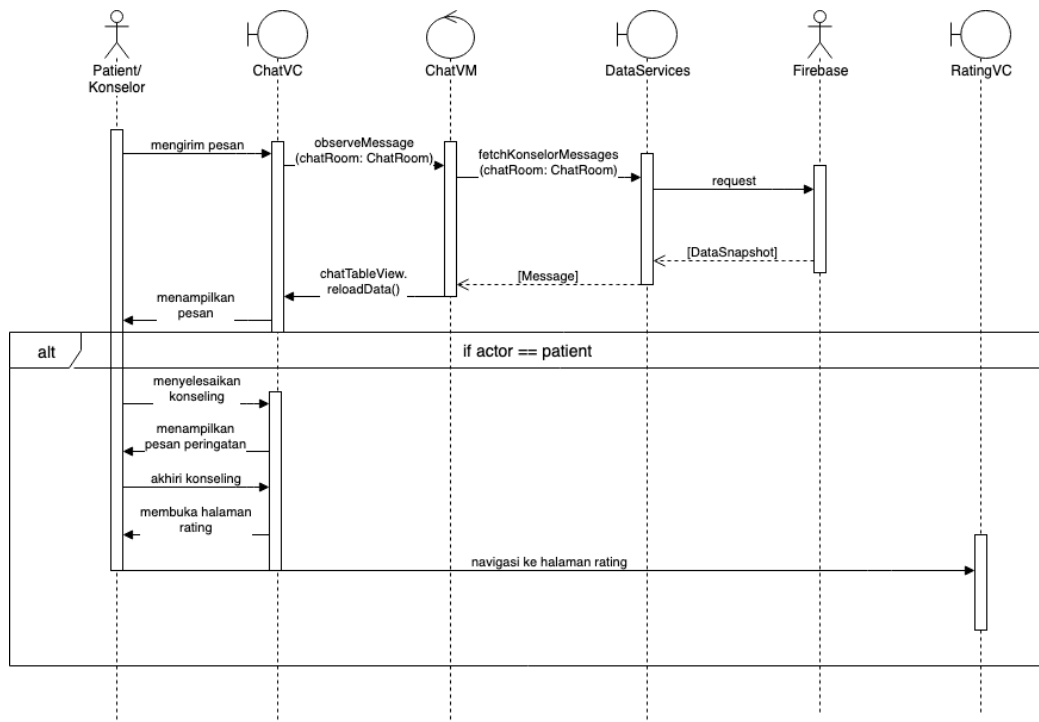


Gambar 5.7 Sequence Diagram - Mengirim Permintaan Konseling

5.2.7 Sequence Diagram - Melakukan Konseling

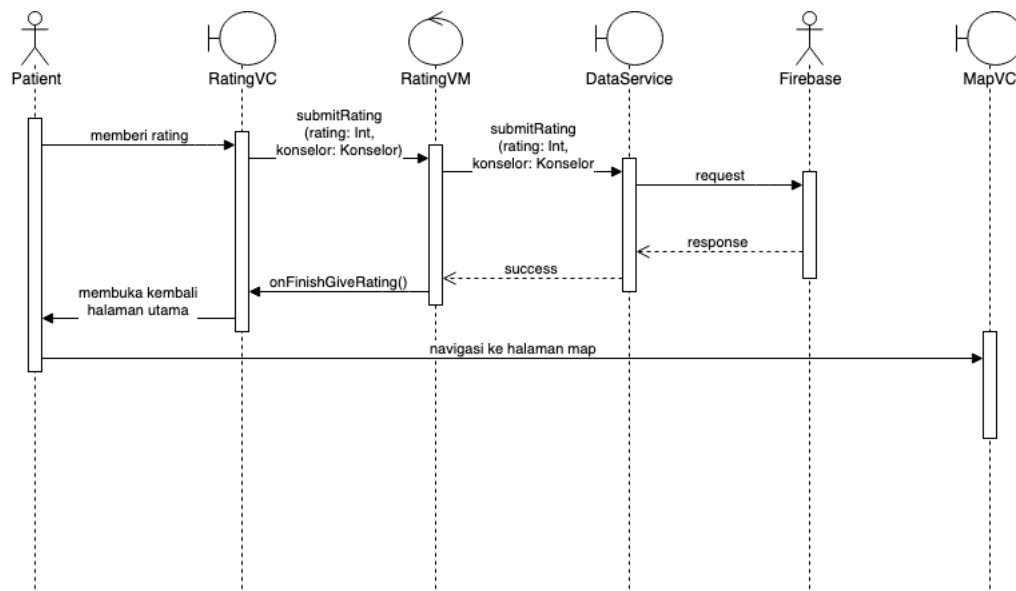
Sequence diagram melakukan konseling oleh aktor pasien digambarkan pada Gambar 5.8. Aktor pasien mengirim pesan pada tempat yang disediakan pada halaman *ChatVC* yang kemudian memanggil fungsi `observeMessage(chatRoom: ChatRoom)` pada objek *ChatVM*. Kemudian objek *ChatVM* memanggil fungsi `fetchKonselorMessages(chatRoom: ChatRoom)` pada objek *DataServices*. Objek *DataService* kemudian melakukan *request* kepada *Firestore* dan mendapatkan respon berupa *array* bertipe data *Message*. Kemudian objek *ChatVM* melakukan *trigger* pada *chatTableView* untuk memanggil fungsi `reloadData()` yang berguna untuk menambahkan pesan kepada halaman *ChatVC*. Setelah menampilkan pesan maka aktor pasien dapat menyelesaikan konseling dengan menekan tombol selesai pada halaman *ChatVC*. Kemudian *ChatVC* menampilkan pesan pop-up untuk mengonfirmasi aksi dari aktor pasien.

Kemudian aktor pasien menekan tombol akhiri konseling dan sistem membuka halaman rating yaitu *RatingVC*.



Gambar 5.8 Sequence Diagram - Melakukan Konseling

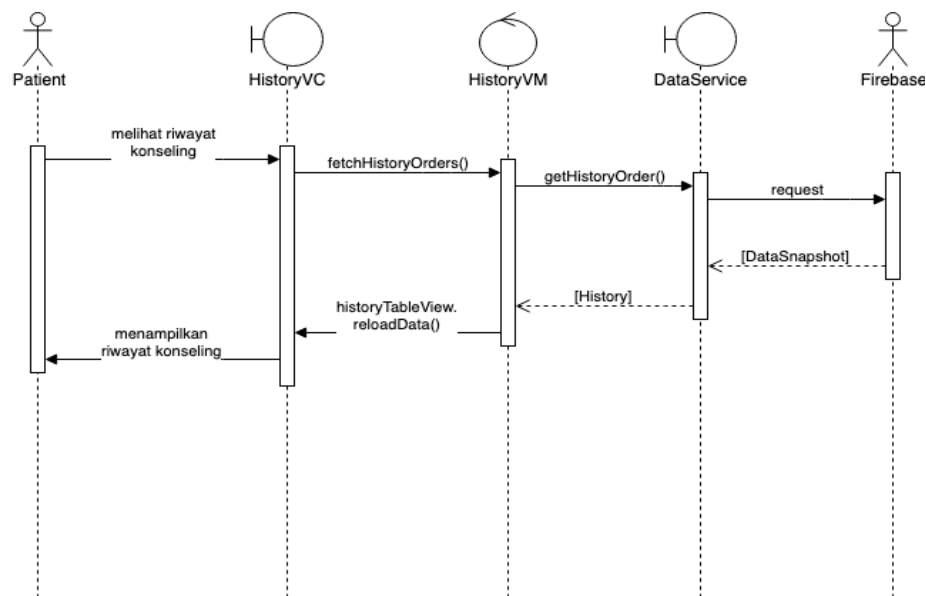
5.2.8 Sequence Diagram - Memberi Rating ke Konselor



Gambar 5.9 Sequence Diagram - Memberi Rating ke Konselor

Sequence diagram memberi rating ke konselor oleh aktor pasien digambarkan pada Gambar 5.9. Setelah selesai melakukan konseling dengan konselor, maka pasien dapat memberi rating pada tempat yang disediakan pada halaman *RatingVC*. Kemudian memanggil fungsi `submitRating(rating: Int, konselor: Konselor)` yang ada pada objek *RatingVM*. Kemudian objek *RatingVM* memanggil fungsi `submitRating(rating: Int, konselor: Konselor)` pada objek *DataService*. Objek *DataService* melakukan *request* ke Firebase dan mendapatkan respon bahwa pasien sukses memberi rating ke konselor. Kemudian objek *RatingVM* memanggil fungsi `onFinishGiveRating()` pada *RatingVC* dan sistem navigasi ke halaman *MapVC*.

5.2.9 Sequence Diagram - Melihat Riwayat Konseling

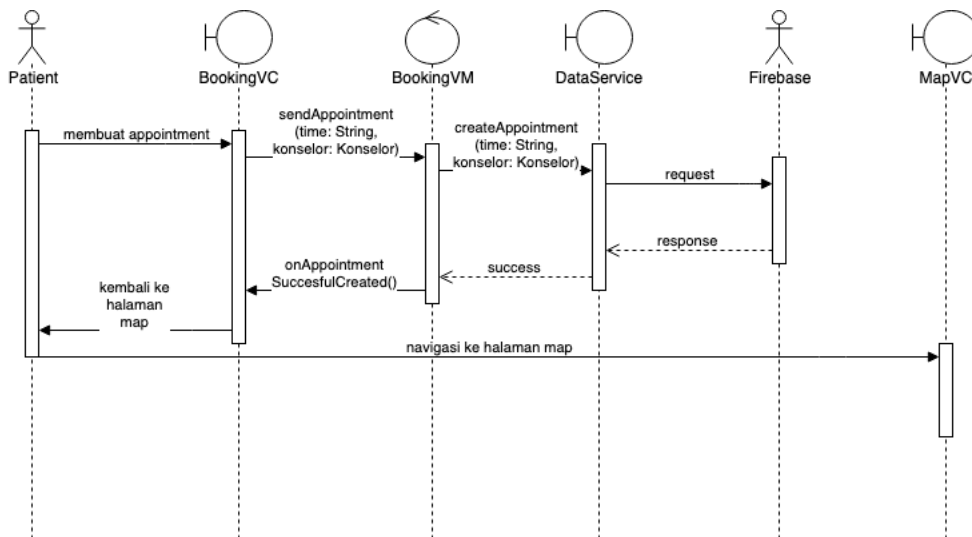


Gambar 5.10 Sequence Diagram - Melihat Riwayat Konseling

Sequence diagram melihat riwayat konseling dari aktor pasien digambarkan pada Gambar 5.10. Aktor pasien melihat riwayat konseling pada halaman *HistoryVC* yang kemudian memanggil fungsi `fetchHistoryOrders()` dari objek *HistoryVM*. Kemudian objek *HistoryVM* memanggil fungsi `getHistoryOrder()` pada objek *DataService*. Setelah itu objek *DataService* melakukan *request* ke Firebase dan mendapatkan respon berupa *array* bertipe data *History*. Kemudian objek *HistoryVM* melakukan *trigger* kepada *historyTableView* untuk memanggil fungsi `reloadData()` agar halaman *HistoryVC* menampilkan data riwayat konseling kepada pasien.

5.2.10 Sequence Diagram - Membuat Appointment

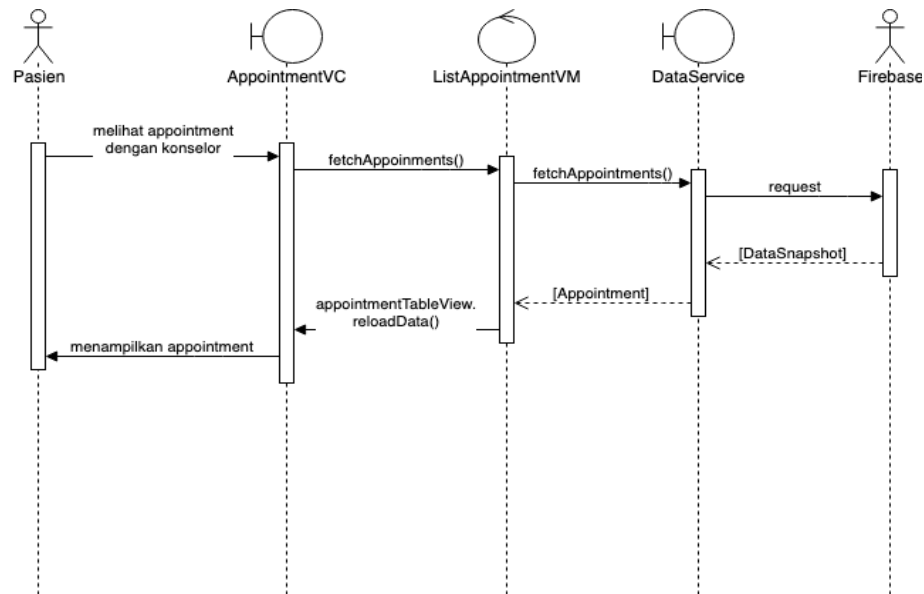
Sequence diagram membuat *appointment* dari aktor pasien digambarkan pada Gambar 5.11. Aktor pasien membuat *appointment* pada halaman *BookingVC* dan memilih jam yang tersedia pada hari ini. Kemudian dari *BookingVC* memanggil fungsi `sendAppointment(time: String, konselor: Konselor)` pada objek *BookingVM*. Kemudian objek *BookingVM* memanggil fungsi `createAppointment(time: String, konselor: Konselor)` pada objek *DataService*. Setelah itu objek *DataService* melakukan *request* ke *Firestore* dan mendapatkan respon bahwa *appointment* berhasil dibuat sehingga objek *BookingVM* memanggil fungsi `onAppointmentSuccessfulCreated()` pada *BookingVC*. Kemudian sistem kembali menampilkan halaman *MapVC* sebagai halaman utama.



Gambar 5.11 Sequence Diagram - Membuat Appointment

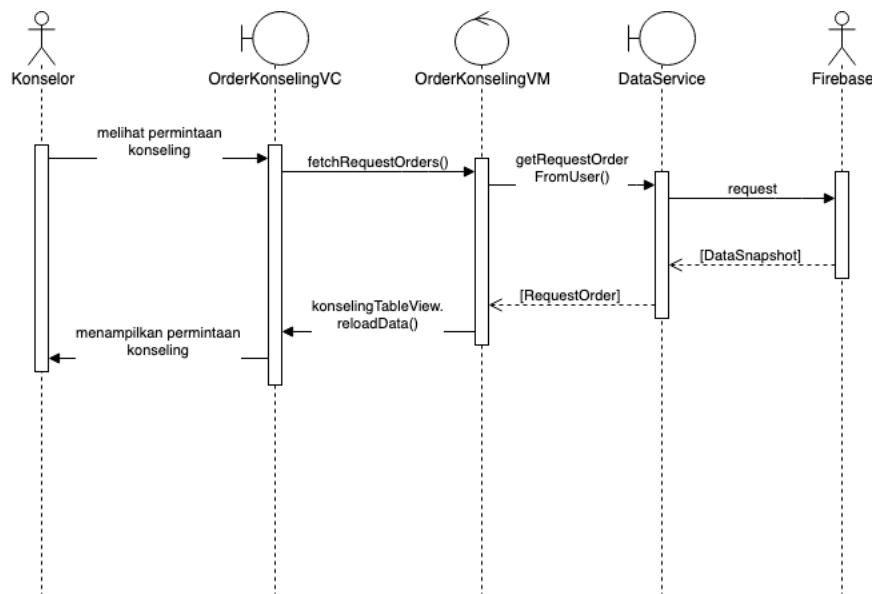
5.2.11 Sequence Diagram - Melihat Daftar Appointment Dengan Konselor

Sequence diagram melihat daftar *appointment* dengan konselor digambarkan pada Gambar 5.12. Aktor pasien melihat daftar *appointment* pada halaman *AppointmentVC* yang kemudian memanggil fungsi `fetchAppointments()` dari objek *ListAppointmentVM*. Kemudian objek *ListAppointmentVM* memanggil fungsi `fetchAppointments()` pada objek *DataService*. Setelah itu objek *DataService* melakukan *request* ke *Firestore* dan mendapatkan respon berupa *array* bertipe data *Appointment*. Kemudian objek *ListAppointmentVM* melakukan *trigger* kepada *appointmentTableView* untuk memanggil fungsi `reloadData()` agar halaman *AppointmentVC* menampilkan data riwayat konseling kepada pasien.



Gambar 5.12 Sequence Diagram - Melihat Daftar Appointment Dengan Konselor

5.2.12 Sequence Diagram - Melihat Daftar Permintaan Konseling

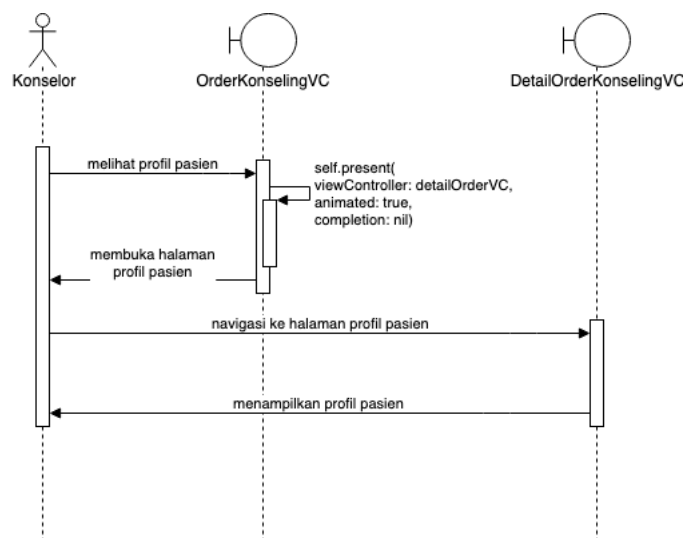


Gambar 5.13 Sequence Diagram - Melihat Daftar Permintaan Konseling

Sequence diagram melihat daftar permintaan konseling oleh aktor konselor digambarkan pada Gambar 5.13. Aktor konselor melihat daftar permintaan konseling pada halaman *OrderKonselingVC* yang kemudian memanggil fungsi *fetchRequestOrders()* dari objek *OrderKonselingVM*. Kemudian objek *OrderKonselingVM* memanggil fungsi *getRequestOrderFromUser()* pada objek *DataService*. Setelah itu objek *DataService* melakukan *request* ke Firebase dan mendapatkan respon berupa *array* bertipe data *RequestOrder*. Kemudian objek *OrderKonselingVM* melakukan *trigger* kepada *konselingTableView* untuk memanggil fungsi *reloadData()* agar halaman *OrderKonselingVC* menampilkan data riwayat konseling kepada pasien.

5.2.13 Sequence Diagram - Melihat Detail Profil Pasien

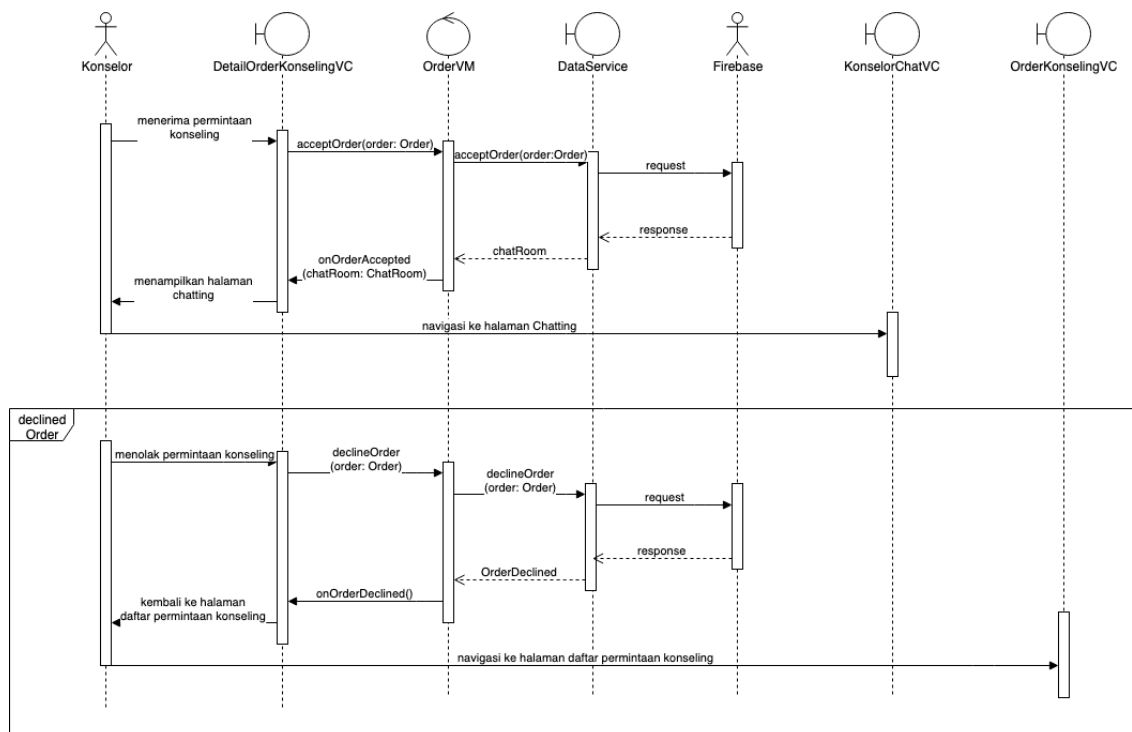
Sequence diagram melihat profil pasien oleh aktor konselor digambarkan pada Gambar 5.14. Pada halaman *OrderKonselingVC*, aktor konselor memilih pasien yang tersedia untuk dapat melihat detail profilnya. Kemudian pada sistem membuka halaman profil pasien yang bernama *DetailOrderKonselingVC*.



Gambar 5.14 Sequence Diagram - Melihat Profil Pasien

5.2.14 Sequence Diagram - Merespon Permintaan Konseling

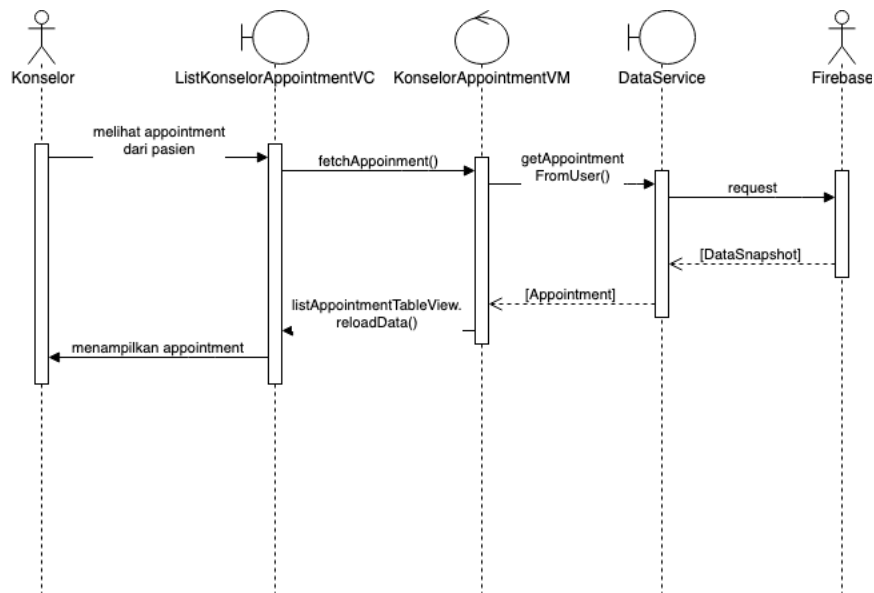
Sequence diagram merespon permintaan konseling dari aktor konselor digambarkan pada Gambar 5.15. Konselor menekan tombol terima konseling yang disediakan oleh *DetailOrderKonselingVC* dan memanggil fungsi `acceptOrder(order: Order)` dari objek *OrderVM*. Kemudian objek *OrderVM* memanggil fungsi `acceptOrder(order: Order)` dari objek *DataService*. Setelah itu objek *DataService* melakukan *request* ke Firebase dan mendapatkan respon berupa objek *chatRoom*. Kemudian objek *OrderVM* memanggil fungsi `onOrderAccepted(chatRoom: ChatRoom)` dan menampilkan halaman *KonselorChatVC*.



Gambar 5.15 Sequence Diagram - Merespon Permintaan Konseling

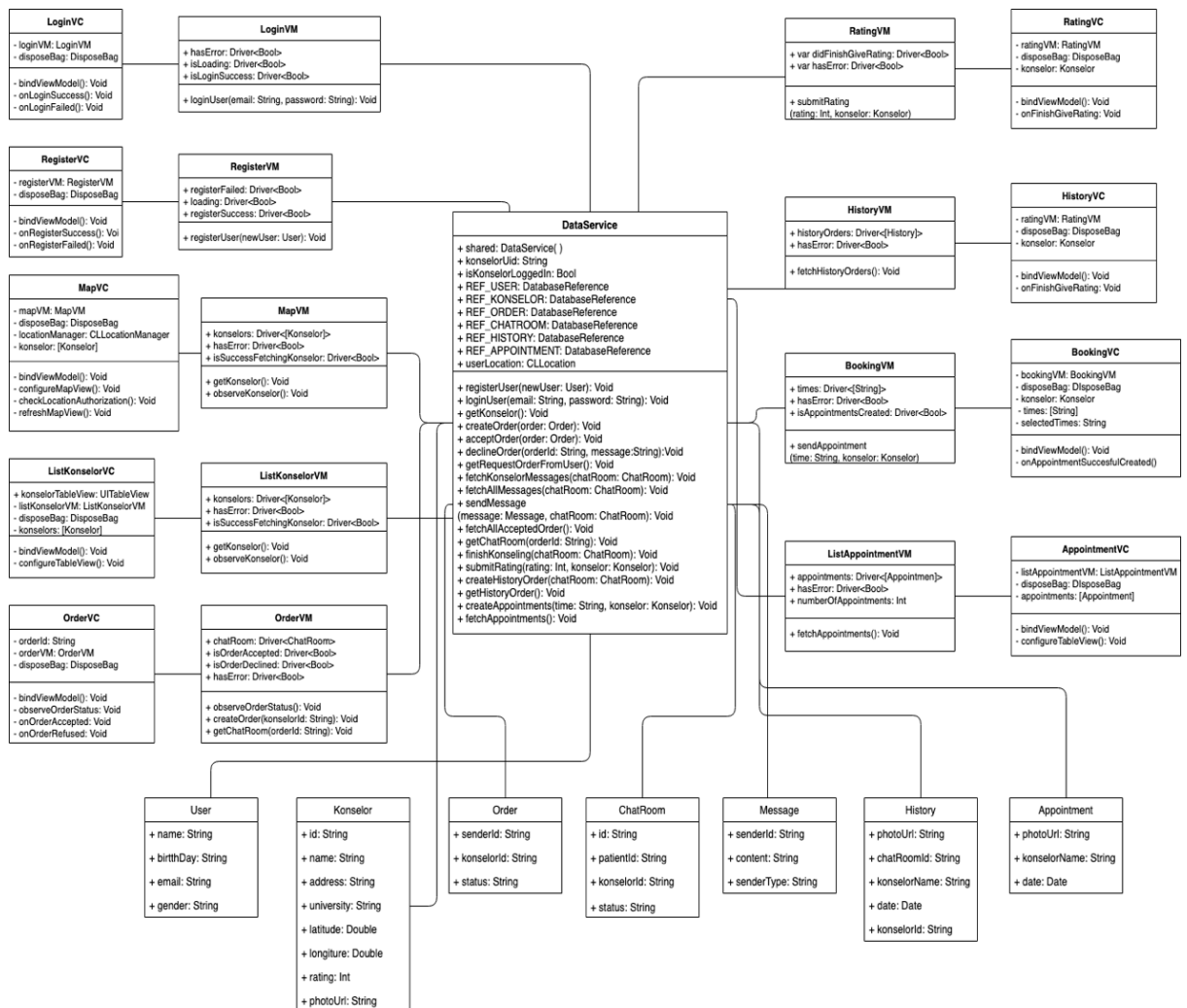
5.2.15 Melihat Daftar Appointment Dari Pasien

Sequence diagram melihat daftar *appointment* dari pasien digambarkan pada Gambar 5.16. Aktor konselor melihat daftar *appointment* pada halaman *ListKonselorAppointmentVC* yang kemudian memanggil fungsi *fetchAppointments()* dari objek *KonselorAppointmentVM*. Kemudian objek *KonselorAppointmentVM* memanggil fungsi *getAppointmentFromUser()* pada objek *DataService*. Setelah itu objek *DataService* melakukan *request* ke *Firebase* dan mendapatkan respon berupa *array* bertipe data *Appointment*. Kemudian objek *KonselorAppointmentVM* melakukan *trigger* kepada *listAppointmentTableView* untuk memanggil fungsi *reloadData()* agar halaman *ListKonselorAppointmentVC* menampilkan data riwayat konseling kepada pasien.

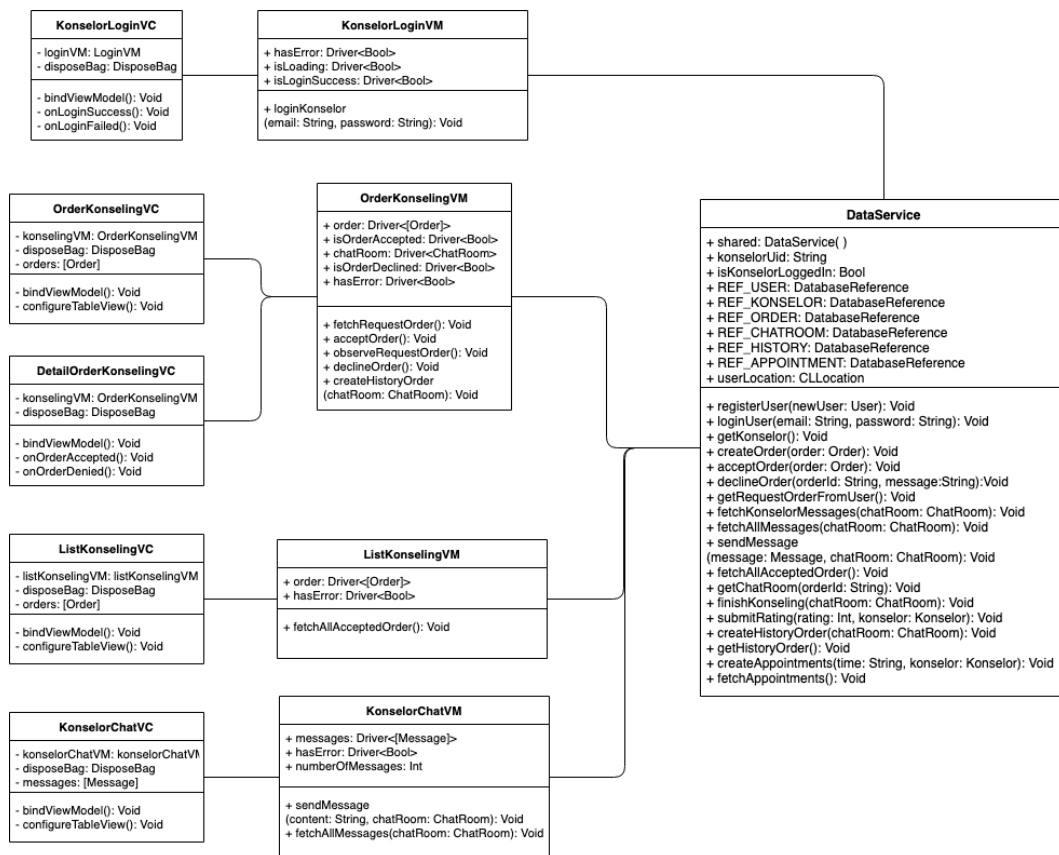


Gambar 5.16 Sequence Diagram - Melihat Daftar Appointment Dari Pasien

5.3 Class Diagram



Gambar 5.17 Class Diagram Aplikasi Pasien



Gambar 5.18 Class Diagram - Aplikasi Sisi Konselor

Class Diagram adalah diagram yang mendeskripsikan arsitektur objek dan class pada sistem beserta hubungan keterkaitan antar satu dengan yang lain dan menjelaskan bagaimana tiap modul dikelompokkan. *Class Diagram* yang dibuat penelitian ini dikelompokkan menjadi 2, yaitu *Class Diagram* untuk aplikasi sisi pasien dan *Class Diagram* untuk aplikasi sisi konselor. *Class Diagram* untuk pasien digambarkan pada Gambar 5.17 dan *Class Diagram* untuk konselor digambarkan pada Gambar 5.18. Penjelasan tiap class pada *Class Diagram* aktor pasien dijelaskan pada Tabel 5.1 dan penjelasan tiap class pada *Class Diagram* aktor konselor dijelaskan pada Tabel 5.2.

Tabel 5.1 Penjelasan Class Diagram Pasien

No	Class	Responsibility
1.	LoginVC	Menyediakan tempat bagi pasien untuk login ke sistem dan memonitor validasi login pada fungsi <code>onLoginSuccesfull</code> dan <code>onLoginFailed</code>
2.	LoginVM	Mengatur <i>business-logic</i> untuk login pada class <code>LoginVC</code> dengan fungsi <code>loginUser</code>

3.	RegisterVC	Menyediakan tempat bagi pasien untuk register ke sistem dan memonitor validasi register pada fungsi <code>onRegisterSuccesfull</code> dan <code>onRegisterFailed</code>
4.	RegisterVM	Mengatur <i>business-logic</i> untuk register pada class <i>RegisterVC</i> dengan fungsi <code>registerUser</code>
5.	MapVC	Menampilkan lokasi konselor yang terdekat dengan lokasi pasien pada fungsi <code>configureMapView</code>
6.	MapVM	Mengatur <i>business-logic</i> untuk menampilkan lokasi konselor pada class <i>MapVC</i> dengan fungsi <code>observeKonselor</code>
7.	ListKonselorVC	Menampilkan daftar konselor terdekat dengan pasien pada fungsi <code>configureTableView</code>
8.	ListKonselorVM	Mengatur <i>business-logic</i> untuk menampilkan daftar konselor terdekat dengan pasien pada class <i>ListKonselorVM</i> dengan fungsi <code>observeKonselor</code>
9.	OrderVC	Menampilkan halaman menunggu konfirmasi permintaan konseling dari pasien ke konselor pada fungsi <code>observeOrderStatus</code>
10.	OrderVM	Mengatur <i>business-logic</i> memonitor status permintaan konseling pada class <i>OrderVC</i> dengan fungsi <code>observeOrderStatus</code>
11.	RatingVC	Menampilkan halaman rating setelah pasien selesai melakukan konseling. Pada class ini menyediakan tempat bagi pasien untuk memberi rating pada konselor pada fungsi <code>onFinishGiveRating</code>
12.	RatingVM	Mengatur <i>business-logic</i> untuk pasien dapat memberi rating pada class <i>RatingVC</i> dengan fungsi <code>submitRating</code>
13.	HistoryVC	Menampilkan halaman riwayat konseling antara pasien dengan konselor pada fungsi <code>fetchHistoryOrders</code>
14.	HistoryVM	Mengatur <i>business-logic</i> untuk menampilkan riwayat konseling antara pasien dengan konselor pada class <i>HistoryVC</i> dengan fungsi <code>fetchHistoryOrders</code>
15.	BookingVC	Menyediakan tempat bagi pasien untuk membuat <i>appointment</i> dengan konselor pada fungsi <code>onAppointmentSuccesfulCreated</code>
16.	BookingVM	Mengatur <i>business-logic</i> untuk pasien dapat membuat

		<i>appointment</i> dengan konselor pada <i>class BookingVC</i> dengan fungsi <code>sendAppointment</code>
17.	AppointmentVC	Menampilkan daftar <i>appointment</i> pasien dengan konselor pada fungsi <code>configureTableView</code>
18.	AppointmentVM	Mengatur <i>business-logic</i> untuk pasien menampilkan daftar <i>appointment</i> pada <i>class AppointmentVC</i> dengan fungsi <code>fetchAppointment</code>
19.	User	Merupakan representasi objek dari pasien
20.	Konselor	Merupakan representasi objek dari konselor
21.	Order	Merupakan representasi objek order yaitu transaksi konseling antara pasien dengan konselor.
22.	ChatRoom	Merupakan representasi objek untuk ruang percakapan untuk konseling antara pasien dengan konselor.
23.	Message	Merupakan representasi objek dari pesan.
24.	History	Merupakan representasi objek dari riwayat konseling pada pasien.
25.	Appointment	Merupakan representasi objek dari <i>appointment</i> yang dibuat pasien.

Tabel 5.2 Penjelasan Class Diagram Konselor

No	Class	Responsibility
1.	KonselorLoginVC	Menyediakan tempat bagi konselor untuk login ke sistem dan memonitor validasi login pada fungsi <code>onLoginSuccesfull</code> dan <code>onLoginFailed</code>
2.	KonselorLoginVM	Mengatur <i>business-logic</i> untuk login pada <i>class LoginVC</i> dengan fungsi <code>loginKonselor</code>
3.	OrderKonselingVC	Menampilkan daftar permintaan konseling dari pasien pada fungsi <code>configureTableView</code>
4.	DetailOrderKonselingVC	Menampilkan detail profil pasien yang mengirim permintaan konseling dan memonitor apakah konselor menerima atau menolak konseling pada fungsi <code>onOrderAccepted</code> dan <code>onOrderDenied</code> .
5.	OrderKonselingVM	Mengatur <i>business-logic</i> pada <i>class OrderKonselingVC</i> dan <i>DetailOrderKonselingVC</i> untuk memonitor status dari order pada

		<code>observeOrderStatus.</code>
6.	ListKonselingVC	Menampilkan daftar konseling dengan pasien pada fungsi <code>configureTableView</code>
7.	ListKonselingVM	Mengatur <i>business-logic</i> pada <i>class ListKonselingVC</i> untuk memperoleh data konseling pada fungsi <code>fetchAllAcceptedOrder</code>
8.	KonselorChatVC	Menampilkan halaman chat antara konselor dengan pasien pada fungsi <code>configureTableView</code>
9.	KonselorChatVM	Mengatur <i>business-logic</i> pada <i>class KonselorChatVM</i> untuk memperoleh data pesan antara konselor dan pasien pada fungsi <code>fetchAllMessages</code>

5.4 Perancangan Struktur Data

Perancangan struktur data dibuat untuk merepresentasikan entitas objek yang digunakan dalam pengembangan aplikasi yang dilakukan. Proses penyimpanan data yang dilakukan pada aplikasi yang dikembangkan menggunakan layanan Firebase *Realtime Database*. Entitas objek yang didefinisikan terdiri dari *Patient*, *Konselor*, *Order*, *Appointment*, dan *ChatRoom*. Berikut merupakan hasil perancangan dari tiap entitas yang telah didefinisikan.

Tabel 5.3 Perancangan Data Patient

No	Nama Field	Deskripsi
1.	name	Merupakan nama lengkap pasien
2.	email	Merupakan alamat email dari pasien
3.	gender	Merupakan jenis kelamin dari pasien
4.	birthday	Merupakan tanggal lahir dari pasien
5.	latitude	Merupakan satuan posisi latitude dari pasien
6.	longitude	Merupakan satuan posisi longitude dari pasien

Tabel 5.4 Perancangan Data Konselor

No	Nama Field	Deskripsi
1.	name	Merupakan nama lengkap konselor
2.	email	Merupakan alamat email dari konselor
3.	address	Merupakan alamat dari pasien

4.	university	Merupakan nama universitas tempat konselor menempuh studinya
5.	photoUrl	Merupakan link url foto profil dari konselor
6.	patient_count	Merupakan jumlah total pasien yang pernah konselor layani dalam aplikasi
7.	rating	Merupakan rating dari konselor
8.	schedule	Merupakan objek berisi jadwal tersedia dari konselor

Tabel 5.5 Perancangan Data Order

No	Nama Field	Deskripsi
1.	senderId	Merupakan id pasien dalam permintaan konseling
2.	konselorId	Merupakan id konselor dalam permintaan konseling
3.	status	Merupakan keterangan order

Tabel 5.6 Perancangan Data Appointment

No	Nama Field	Deskripsi
1.	date	Merupakan keterangan waktu dari <i>appointment</i>
2.	konselorName	Merupakan nama konselor dalam <i>appointment</i>
3.	patientId	Merupakan id pasien dalam <i>appointment</i>
4.	photoUrl	Merupakan link url dari foto profil konselor

Tabel 5.7 Perancangan Data ChatRoom

No	Nama Field	Deskripsi
1.	patientId	Merupakan id pasien dalam permintaan konseling
2.	konselorId	Merupakan id konselor dalam permintaan konseling
3.	status	Merupakan keterangan kondisi dari sesi konseling
4.	messages	Merupakan pesan antara pasien dengan konselor dalam sesi konseling

5.5 Perancangan Algoritme

Perancangan algoritme dilakukan untuk membuat prosedur yang berisi langkah-langkah yang diperlukan dalam menyelesaikan sebuah permasalahan. Dalam aplikasi yang dikembangkan pada penelitian ini, terdapat 3 algoritme fitur utama pada aplikasi yaitu memperoleh konselor terdekat dengan pasien yang dijelaskan pada Tabel 5.8, mengirim permintaan konseling yang dijelaskan pada Tabel 5.9, dan membuat *appointment* dengan konselor yang dijelaskan pada Tabel 5.10.

Tabel 5.8 Perancangan Algoritme - Memperoleh Data Konselor Terdekat

1	func getKonselor(completion: (_ konselor: [Konselor]-> Void) {
2	input: -
3	output: [Konselor]
4	var konselorArray: [Konselor]()
5	REF_KONSELOR.observeSingleEvent() { (dataSnapshot) in
6	var rawKonselor = dataSnapshot as [DataSnapshot]
7	for konselor in rawKonselor {
8	let newKonselor = Konselor(konselor: konselor)
9	let latitude = newKonselor.latitude
10	let longitude = newKonselor.longitude
11	newKonselor.distance = calculateDistance(latitude,
12	longitude)
13	konselorArray.append(newKonselor)
14	}
15	konselorArray.sorted(by: \$0.distance < \$1.distance)
16	completion(konselorArray)
17	}
18	
19	func calculateDistance(latitude: Double, longitude: Double) ->
20	Double {
21	var userCoordinate = CLLocation()
22	var konselorCoordinate = CLLocation(latitude: latitude,
23	longitude: longitude)
24	return userCoordinate.distance(from: konselorCoordinate) / 1000
25	}

Tabel 5.9 Perancangan Algoritme - Mengirim Permintaan Konseling

1	func createOrder(order: Order) {
2	input: Order
3	ouput: Bool
4	
5	let newOrder = Dictionary<String,String>()
6	let newOrderId = REF_ORDER.childByAutoId().key
7	REF_ORDER.child(newOrderId).updateChild(newOrder)
8	completion(true)
9	}

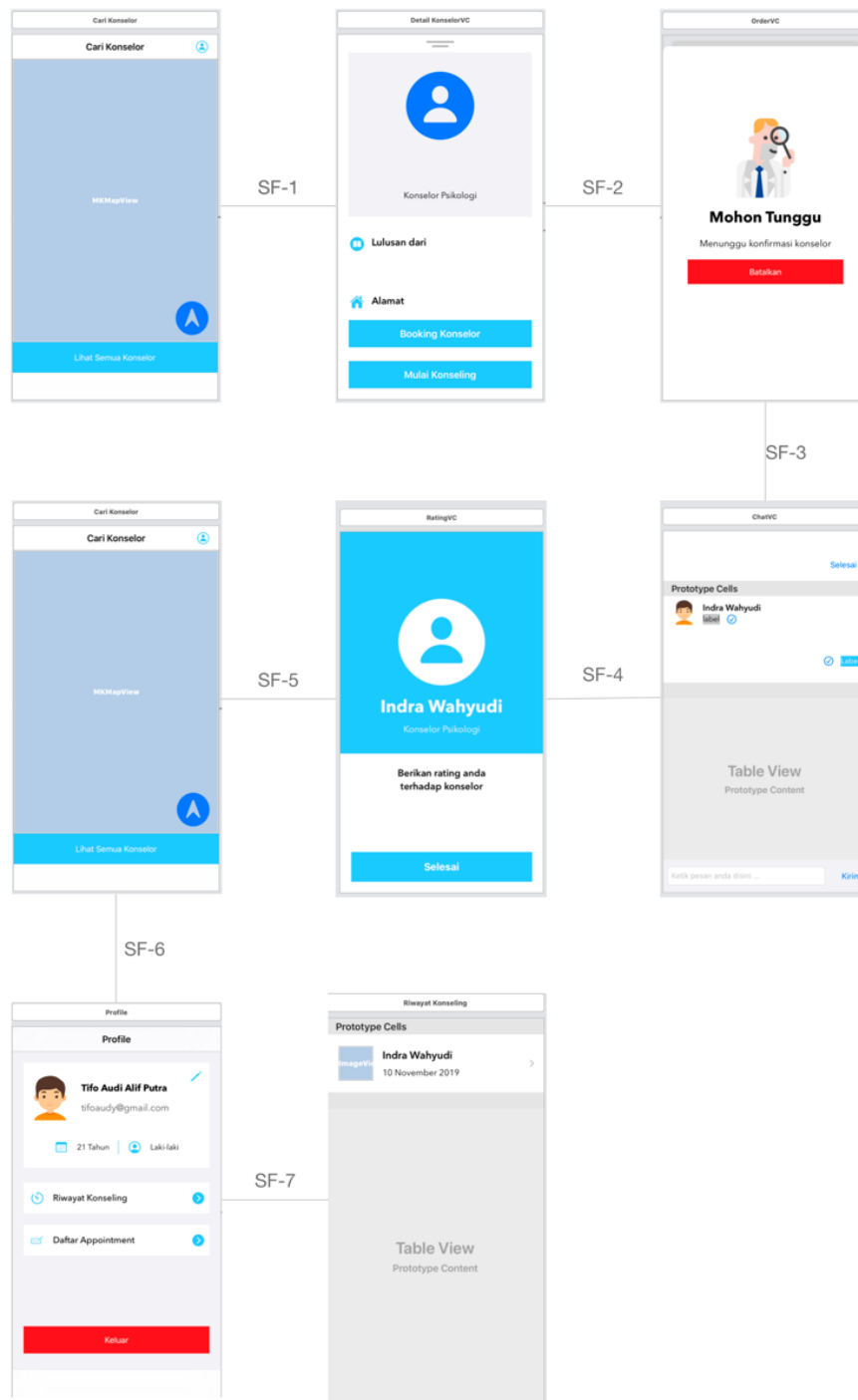
Tabel 5.10 Perancangan Algoritme - Membuat Appointment

1	func createAppointment(time: String, konselor: Konselor) {
2	input: time, konselor
3	output: Bool
4	
5	let uid = Auth.auth().currentUser.uid
6	let newAppointment = Dictionary<String,String>()
7	REF_APPOINTMENT.childByAutoId.updateChild(newAppointment)
8	completion(true)
9	}

5.6 Perancangan Antarmuka

Screenflow dibuat agar memudahkan dalam memahami alur sistem dalam membangun *user interface*. Perancangan *screenflow* pada penelitian ini menggunakan *storyboard* yang merupakan *tool* dari *Xcode* sebagai IDE untuk pengembangan aplikasi perangkat bergerak pada *platform iOS*. Pada aplikasi sisi pasien, *screenflow* yang dirancang dapat dilihat pada Gambar 5.19 dan *screenflow* aplikasi sisi konselor yang dirancang dapat dilihat pada Gambar 5.20.

5.6.1 Screenflow - Aplikasi Sisi Pasien

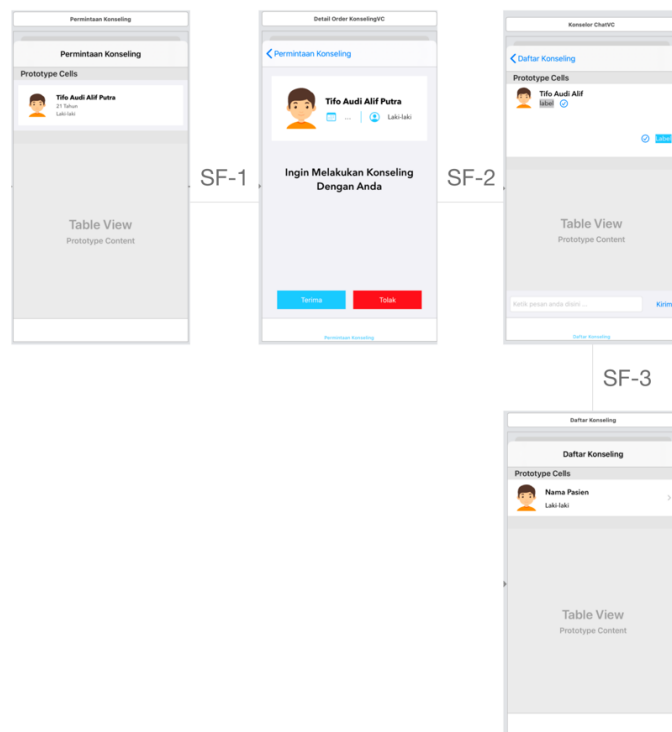


Gambar 5.19 Screenflow Aplikasi Pasien

Tabel 5.11 Penjelasan Screenflow Pasien

No.	Flow Identifier	Penjelasan
1.	SF-1	Pasien memilih konselor dan menampilkan profil konselor.
2.	SF-2	Pasien mengirim permintaan konseling dan menampilkan halaman menunggu konfirmasi konselor.
3.	SF-3	Konselor menerima permintaan dari pasien dan menampilkan halaman <i>chat</i> .
4.	SF-4	Pasien telah selesai melakukan konseling dan menampilkan halaman <i>rating</i> .
5.	SF-5	Pasien telah selesai memberi rating ke konselor dan kembali ke halaman peta.
6.	SF-6	Dari halaman peta, pasien menampilkan halaman profil
7.	SF-7	Pasien menampilkan halaman riwayat konseling.

5.6.2 Screenflow - Aplikasi Sisi Konselor



Gambar 5.20 Screenflow Aplikasi Konselor

Tabel 5.12 Penjelasan Screenflow Konselor

No	Flow Identifier	Penjelasan
1	SF-1	Konselor melihat daftar permintaan konseling dari pasien dan menampilkan profil pasien
2	SF-2	Konselor menerima permintaan konseling dari pasien dan menampilkan halaman <i>chat</i>
3	SF-3	Sesi konseling telah selesai dan sistem menampilkan halaman awal pada sistem

5.7 Implementasi Perangkat Lunak

Tahap implementasi perangkat lunak menjelaskan tentang proses implementasi untuk mengembangkan aplikasi dari hasil perancangan yang telah selesai dilakukan pada tahap sebelumnya. Pembahasan mengenai tahap implementasi meliputi spesifikasi lingkungan sistem dalam tahap implementasi, batasan pada tahap implementasi, implementasi perancangan struktur data, implementasi kode program, dan implementasi antarmuka.

5.7.1 Spesifikasi Lingkungan Sistem

5.7.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan pada penelitian ini dijelaskan pada Tabel 5.13.

Tabel 5.13 Spesifikasi Perangkat Keras

<i>Processor</i>	3,1 GHz Dual-Core Intel Core i5
<i>Ram</i>	8 GB 2133 MHz LPDDR3
<i>GPU</i>	Intel Iris Plus Graphics 650 1536 MB
<i>Storage</i>	256GB PCI-r based flash storage
<i>OS</i>	macOS Catalina 10.15

5.7.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada penelitian ini dijelaskan pada Tabel 5.14.

Tabel 5.14 Spesifikasi Perangkat Lunak

IDE	Xcode 11.1
<i>Design Tools</i>	<i>Draw.io</i> dan <i>Storyboard</i>
<i>Document</i>	Microsoft Office Word for Mac Ver. 16.16
<i>Programming Language</i>	Swift 5.1
<i>Device Testing</i>	iPhone 7

5.7.2 Batasan Implementasi

Batasan implementasi yang ditetapkan pada penelitian ini adalah sebagai berikut:

1. Aplikasi dibangun pada platform iOS dengan spesifikasi OS minimal iOS 13.
2. Sistem basis data yang digunakan menggunakan layanan *Realtime Database* milik Firebase.
3. Arsitektur sistem yang digunakan adalah MVVM (*Model-View-ViewModel*)

5.7.3 Implementasi Struktur Data

Implementasi struktur data dilakukan dengan menggunakan *realtime database* milik Firebase. Struktur data yang digunakan pada layanan *realtime database* adalah *tree*. Hasil implementasi struktur data *tree* terdiri dari *appointment*, *chat_room*, *konselor*, *order*, dan *user* yang masing-masing memiliki *child tree* sebagai properti atau atribut. Hasil implementasi tabel data pada aplikasi yang dibangun digambarkan secara jelas pada Gambar 5.21.



Gambar 5.21 Implementasi Struktur Data

5.7.4 Implementasi Kode Program

Implementasi kode program pada penelitian ini menggunakan bahasa *Swift* yang merupakan bahasa resmi untuk pemrograman aplikasi perangkat bergerak pada sistem operasi iOS. Implementasi dilakukan atas dasar hasil perancangan algoritme pada sub bab sebelumnya. Berikut merupakan implementasi kode program yang dikembangkan.

Kode Program: getKonselor()	
1	func getKonselor(success: @escaping (_ konselor: [Konselor]) ->
2	()), failure: @escaping ()-> ()) {
3	
4	var konselorArray = [Konselor]()
5	REF_KONSELOR.observeSingleEvent(of: .value) { [unowned self]
6	(dataSnapshot) in
7	guard let rawKonselor = dataSnapshot.children.allObjects as?
8	[DataSnapshot] else {
9	failure()
10	return
11	}
12	for konselor in rawKonselor {
13	let name = konselor.childSnapshot(forPath: "name").value as!
14	String
15	let address = konselor.childSnapshot(forPath:
16	"address").value as! String
17	let university = konselor.childSnapshot(forPath:
18	"university").value as! String
19	let latitude = konselor.childSnapshot(forPath:
20	"location").childSnapshot(forPath: "latitude").value as! Double
21	let longitude = konselor.childSnapshot(forPath:
22	"location").childSnapshot(forPath: "longitude").value as! Double
23	let isOnline = konselor.childSnapshot(forPath:
24	"isOnline").value as! Bool
25	let distance = self.calculateDistance(konselor: latitude,
26	konselor: longitude)
27	let rating = konselor.childSnapshot(forPath: "rating").value
28	as! Double
29	let patientCount = konselor.childSnapshot(forPath:
30	"patient_count").value as! Int
31	let photoUrl = konselor.childSnapshot(forPath:
32	"photoUrl").value as! String
	let mondaySchedule = konselor.childSnapshot(forPath:
	"schedule").childSnapshot(forPath: "Monday").value as! String
	let tuesdaySchedule = konselor.childSnapshot(forPath:
	"schedule").childSnapshot(forPath: "Tuesday").value as! String
	let wednesdaySchedule = konselor.childSnapshot(forPath:
	"schedule").childSnapshot(forPath: "Wednesday").value as! String
	let thursdaySchedule = konselor.childSnapshot(forPath:

```

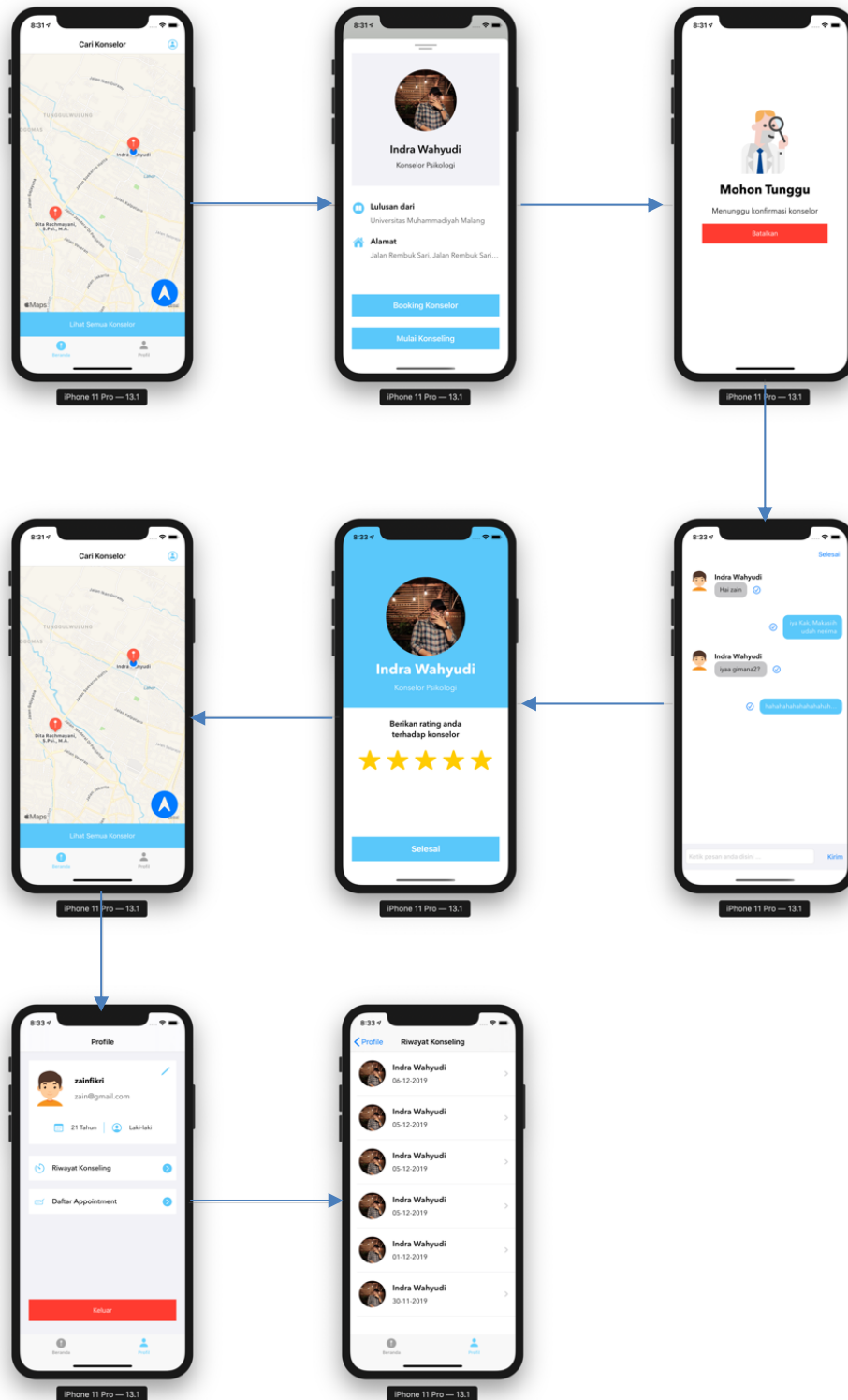
33 "schedule").childSnapshot(forPath: "Thursday").value as! String
34     let fridaySchedule = konselor.childSnapshot(forPath:
35 "schedule").childSnapshot(forPath: "Friday").value as! String
36     let konselorSchedule = KonselorSchedule(monday:
37 mondaySchedule, tuesday: tuesdaySchedule, wednesday:
38 wednesdaySchedule, thursday: thursdaySchedule, friday: fridaySchedule)
39     let newKonselor = Konselor(id: konselor.key, name: name,
40 address: address, university: university, latitude: latitude,
41 longitude: longitude, isOnline: isOnline, distance: distance,
42 patientCount: patientCount, rating: rating, photoUrl: photoUrl,
43 schedule: konselorSchedule)
44
45     if newKonselor.isOnline {
46         konselorArray.append(newKonselor)
47     }
48     konselorArray = konselorArray.sorted(by: { $0.distance <
49 $1.distance })
50     success(konselorArray)
51 }
52
53 func calculateDistance(konselor latitude: Double, konselor
54 longitude: Double) -> Double {
55     guard let userCoordinate = self.userLocation else {
56         return 0
57     }
58     let konselorCoordinate = CLLocation(latitude: latitude,
59 longitude: longitude)
60     let distanceInKm: Double = userCoordinate.distance(from:
61 konselorCoordinate) / 1000
62     return round(100 * distanceInKm)/100
63 }

```

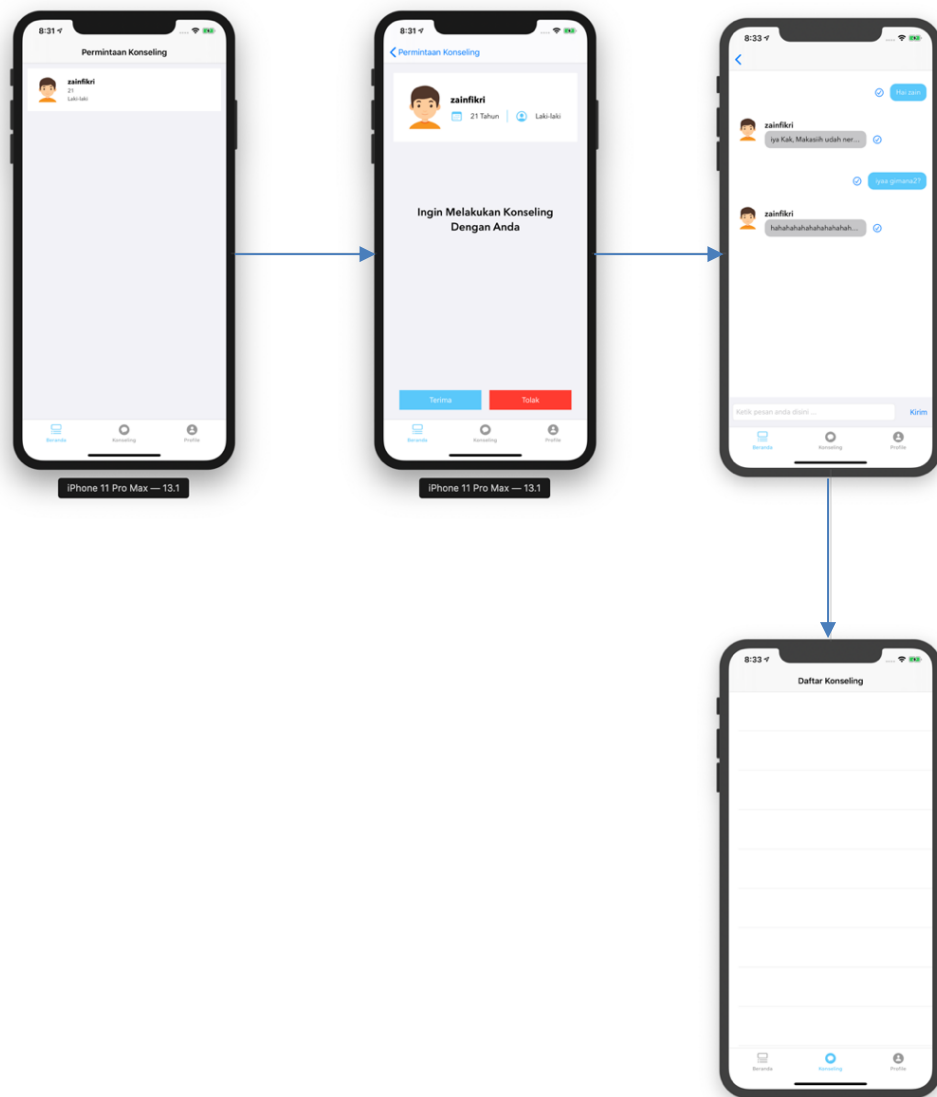
Kode Program: createOrder()	
1	func createOrder(order: Order, completion: @escaping (_ success: Bool, _ orderId: String)->()) {
2	
3	let newOrder = [
4	"senderId": order.senderId,
5	"konselorId": order.konselorId,
6	"status": order.status.rawValue
7]
8	
9	guard let newOrderId = REF_ORDER.childByAutoId().key else {
10	return
11	}
12	
13	REF_ORDER.child(newOrderId).updateChildValues(newOrder)
14	completion(true, newOrderId)
15	}

Kode Program: createAppointment()	
1	func createAppointment(time: String, konselor: Konselor,
2	completion: @escaping(_ success: Bool)->()) {
3	guard let uid = Auth.auth().currentUser?.uid else { return }
4	
5	let newAppointment = [
6	"photoUrl": konselor.photoUrl,
7	"date": time,
8	"patientId": uid,
9	"konselorName": konselor.name
10]
11	REF_APPOINTMENT.childByAutoId().updateChildValues(newAppointment)
12	completion(true)
13	}

5.7.5 Implementasi Antarmuka



Gambar 5.22 Implementasi Antarmuka Pasien



Gambar 5.23 Implementasi Antarmuka Konselor

Implementasi antarmuka merupakan proses mengimplementasikan perancangan antarmuka yang dilakukan dengan menggunakan *storyboard* pada Xcode sebagai *Integrated Development Environment* untuk pengembangan aplikasi perangkat bergerak pada sistem operasi iOS. Implementasi antarmuka dibagi menjadi dua bagian, yaitu implementasi pada aplikasi pasien dan implementasi pada aplikasi konselor. Implementasi antarmuka pada aplikasi pasien digambarkan pada dan implementasi antarmuka pada Gambar 5.22 dan aplikasi konselor digambarkan pada Gambar 5.23.

Alur pada aplikasi sisi pasien dimulai ketika pasien memilih konselor yang tersedia pada halaman peta. Kemudian aplikasi menampilkan detail profil dari konselor yang dipilih oleh pasien. Pasien mengirim permintaan konseling kepada konselor dengan menekan tombol yang telah disediakan. Kemudian aplikasi menampilkan halaman untuk pasien menunggu konfirmasi konselor. Apabila

konselor menerima permintaan tersebut maka aplikasi menampilkan halaman *chatting*. Proses konseling secara online telah terjadi antara pasien dan konselor. Kemudian pasien mengakhiri konseling dengan menekan tombol selesai yang disediakan. Aplikasi akan menampilkan halaman rating untuk pasien memberi rating kepada konselor. Kemudian aplikasi kembali menampilkan halaman peta, pasien dapat melihat riwayat konselingnya dengan membuka *tab profile* terlebih dahulu. Kemudian dari halaman profil, pasien dapat melihat riwayat konselingnya dengan konselor dengan menekan menu yang disediakan.

Alur pada aplikasi sisi konselor dimulai pada halaman untuk menampilkan permintaan konseling. Konselor dapat melihat profil pasien dengan memilih permintaan konseling yang ditampilkan. Apabila konselor menerima permintaan tersebut maka aplikasi menampilkan halaman *chatting*. Jika sesi konseling telah selesai maka aplikasi kembali menampilkan halaman awal.

BAB 6 PENGUJIAN

Pada bab pengujian menjelaskan tentang *blackbox testing* dan pengujian usabilitas pada aplikasi. *Blackbox testing* dilakukan di setiap fitur untuk memastikan aplikasi berjalan dengan sesuai dan memastikan tidak ada *bug*. *Blackbox testing* dilakukan dengan cara menyediakan *input* dan memastikan *output* sesuai dengan harapan. Pengujian usabilitas dilakukan untuk mengukur tingkat kemudahan dalam penggunaan aplikasi.

6.1 Blackbox Testing

6.1.1 Login

Blackbox testing yang dilakukan pada *usecase* login dijelaskan pada Tabel 6.1.

Tabel 6.1 Blackbox testing - Login

<i>Usecase</i>	<i>Target</i>	<i>Test Step</i>	<i>Expected Result</i>	<i>Actual Result</i>
<i>Login</i>	Proses login berhasil.	Mengisi <i>email</i> dan <i>password</i> pada <i>field</i> yang telah disediakan.	Login berhasil dan aplikasi menampilkan halaman utama berupa peta.	Aplikasi berhasil menampilkan halaman utama berupa peta (valid)
	Proses login gagal.	Mengisi <i>email</i> dan <i>password</i> yang tidak valid.	Aplikasi menampilkan pesan bahwa login tidak berhasil.	Aplikasi berhasil menampilkan pesan bahwa login tidak berhasil (valid)

6.1.2 Register

Blackbox testing yang dilakukan pada *usecase* register dijelaskan pada Tabel 6.2.

Tabel 6.2 Blackbox testing - Register

<i>Usecase</i>	<i>Target</i>	<i>Test Step</i>	<i>Expected Result</i>	<i>Actual Result</i>
<i>Register</i>	Proses register berhasil.	Mengisi data diri pada <i>field</i> yang telah disediakan.	Register berhasil dan aplikasi menampilkan halaman utama berupa peta.	Aplikasi berhasil memvalidasi proses register

				pengguna dan menampilkan halaman utama (valid)
	Proses register gagal.	Mengisi data diri yang tidak valid.	Aplikasi menampilkan pesan bahwa register tidak berhasil.	Aplikasi berhasil menampilkan pesan register tidak berhasil (valid)

6.1.3 Melakukan Konseling

Blacbox testing yang dilakukan pada *usecase* melakukan konseling dijelaskan pada Tabel 6.3.

Tabel 6.3 *Blacbox testing* - Melakukan Konseling

<i>Usecase</i>	<i>Target</i>	<i>Test Step</i>	<i>Expected Result</i>	<i>Actual Result</i>
Melakukan Konseling	Pasien dapat melakukan konseling secara online	Pasien memilih konselor dan mengirim permintaan konseling.	Aplikasi menampilkan halaman <i>chatting</i> .	Aplikasi berhasil menampilkan halaman <i>chatting</i> . (valid)
	Konselor menolak permintaan konseling dari pasien	Pasien memilih konselor dan mengirim permintaan konseling.	Aplikasi menampilkan pesan bahwa permintaan konseling ditolak.	Aplikasi berhasil menampilkan pesan bahwa permintaan konseling ditolak (valid)

6.1.4 Membuat Appointment

Blacbox testing yang dilakukan pada *usecase* membuat appoinment dijelaskan pada Tabel 6.4.

Tabel 6.4 *Blacbox testing* - Membuat Appointment

<i>Usecase</i>	<i>Target</i>	<i>Test Step</i>	<i>Expected Result</i>	<i>Actual Result</i>
Membuat Appointment	Proses membuat <i>appointment</i> berhasil	Pasien memilih waktu dari jadwal tersedia konselor dan menekan tombol kirim	Aplikasi menampilkan pesan <i>appointment</i> berhasil dibuat	Aplikasi berhasil menampilkan pesan <i>appointment</i> berhasil dibuat (valid)
	Proses membuat <i>appointment</i> tidak berhasil	Pasien menekan tombol kirim tanpa memilih jadwal yang tersedia	Aplikasi menampilkan pesan untuk memilih waktu yang tersedia dari jadwal konselor.	Aplikasi berhasil menampilkan pesan untuk memilih waktu yang tersedia dari jadwal konselor (valid)

6.2 Pengujian Usabilitas

Pengujian usabilitas merupakan pengujian yang dilakukan untuk mengukur tingkat kemudahan dalam penggunaan sistem. Jumlah responden yang ditentukan pada penelitian ini adalah 5 orang untuk aplikasi sisi pasien dan 5 orang untuk aplikasi sisi konselor. Langkah pengujian usabilitas dimulai ketika responden melakukan semua skenario pengujian yang berisi tugas-tugas yang berbeda dalam mengoperasikan aplikasi. Setelah melakukan semua skenario pengujian, responden diminta untuk mengisi kuisioner untuk memberikan penilaian terhadap aplikasi yang dikembangkan. Instrumen pertanyaan yang digunakan pada pengisian kuisioner menggunakan SUPR-Qm dimana tiap pertanyaan menggunakan skala likert sebagai skala penilaian. Keterangan skala nilai pada skala likert dijelaskan pada Tabel 6.5.

Tabel 6.5 Skala Likert

Nilai	Keterangan
1	Sangat tidak setuju
2	Tidak setuju
3	Netral
4	Setuju
5	Sangat Setuju

Setelah selesai dalam mengisi kuisioner, responden dapat memberikan kritik dan saran terkait aplikasi yang dikembangkan. Kritik dan saran tersebut sangat dibutuhkan untuk menjadi umpan balik pengembang dalam mengembangkan aplikasi lebih baik lagi sehingga dapat dilanjutkan implementasinya pada penelitian lain. Daftar pertanyaan SUPR-Qm dijelaskan pada Tabel 6.6.

Tabel 6.6 Daftar Pertanyaan SUPER-Qm

No	Pertanyaan
1	Aplikasi ini penting untuk saya
2	Aplikasi ini merupakan aplikasi pencari konselor psikologi terbaik yang pernah saya gunakan
3	Saya tidak tahu apakah ada aplikasi pencari konselor psikologi yang lebih baik dari aplikasi ini
4	Saya tidak akan menghapus aplikasi ini dari smartphone saya
5	Saya akan menyarankan aplikasi ini ke teman saya
6	Saya suka mengeksplorasi fitur-fitur yang terdapat dalam fitur ini
7	Aplikasi ini memiliki seluruh fitur dan fungsi yang saya butuhkan dalam mencari konselor psikologi terdekat
8	Saya akan sering membuka aplikasi ini setiap kali saya ingin konseling
9	Aplikasi ini menyenangkan
10	Aplikasi ini bekerja dengan baik dengan fitur-fitur lain yang ada pada smartphone saya (contohnya gps, peta)
11	Saya akan menggunakan aplikasi ini lagi ketika saya ingin mencari konselor untuk konseling
12	Desain dari aplikasi ini memudahkan saya dalam menemukan informasi yang saya inginkan
13	Menurut saya aplikasi ini menarik

14	Kemampuan aplikasi ini sesuai dengan yang saya butuhkan
15	Aplikasi ini sangat mudah dalam hal navigasi
16	Aplikasi ini mudah digunakan

6.2.1 Skenario Pengujian

Skenario pengujian merupakan daftar yang berisi tugas-tugas yang perlu dijalankan ketika mengoperasikan aplikasi untuk melakukan pengujian usability. Daftar skenario pengujian pada aplikasi sisi pasien pada penelitian ini dijelaskan pada Tabel 6.7 dan daftar skenario pengujian pada aplikasi sisi konselor dijelaskan pada Tabel 6.8.

Tabel 6.7 Skenario Pengujian - Aplikasi Sisi Pasien

No	Skenario	Penjelasan
1	Daftar ke dalam aplikasi	Responden diminta untuk membuat akun baru pada aplikasi
2	Login ke aplikasi	Responden diminta login dengan akun yang sudah terdaftar
3	Melihat daftar konselor terdekat	Responden diminta menampilkan halaman daftar konselor terdekat
4	Melihat detail profil konselor	Responden diminta menampilkan halaman detail profil konselor
5	Mengirim permintaan konseling	Responden diminta mengirim permintaan konseling pada konselor
6	Melakukan konseling online dengan konselor	Responden diminta melakukan konseling secara online dengan konselor
7	Memberi rating ke konselor	Responden diminta memberi rating kepada konselor setelah melakukan konseling
8	Membuat appointment dengan konselor	Responden diminta membuat appointment dengan konselor
9	Melihat daftar riwayat konseling	Responden diminta membuka halaman riwayat konseling

Tabel 6.8 Skenario Pengujian - Aplikasi Sisi Konselor

No	Skenario	Penjelasan
1	Login ke aplikasi	Responden diminta login dengan akun yang sudah terdaftar
2	Melihat daftar permintaan konseling	Responden diminta menampilkan permintaan konseling
4	Melihat detail profil pasien	Responden diminta menampilkan halaman detail profil pasien
5	Menerima permintaan konseling	Responden diminta menerima permintaan konseling
6	Menolak permintaan konseling	Responden diminta menolak permintaan konseling
7	Melakukan konseling online dengan pasien	Responden diminta melakukan konseling secara online dengan pasien
8	Melihat daftar appointment dengan pasien	Responden diminta membuka halaman daftar appointment

6.2.2 Hasil Pengujian Usabilitas

Pengujian usabilitas dilakukan oleh 5 responden pada masing-masing aktor yaitu aplikasi pada sisi pasien dan aplikasi pada sisi konselor. Responden melakukan skenario yang telah ditentukan dari masing-masing aplikasi untuk mengukur kemudahan dari aplikasi yang dikembangkan. Hasil pengujian pada skenario yang telah diberikan dari aplikasi sisi pasien dapat dilihat pada Tabel 6.9 sedangkan pada aplikasi sisi konselor dapat dilihat pada Tabel 6.10.

Tabel 6.9 Hasil Pengujian Responden Pasien

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9
User 1	✓	✓	✓	✓	✓	✓	✓	✓	✓
User 2	✓	✓	✓	✓	✓	✓	✓	✓	✓
User 3	✓	✓	✓	✓	✓	✓	✓	✓	✓
User 4	✓	✓	✓	✓	✓	✓	✓	✓	✓
User 5	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabel 6.10 Hasil Pengujian Responden Konselor

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8
User 1	✓	✓	✓	✓	✓	✓	✓	✓
User 2	✓	✓	✓	✓	✓	✓	✓	✓
User 3	✓	✓	✓	✓	✓	✓	✓	✓
User 4	✓	✓	✓	✓	✓	✓	✓	✓
User 5	✓	✓	✓	✓	✓	✓	✓	✓

Dari hasil tersebut, responden telah berhasil menyelesaikan semua tugas yang dibuat pada skenario pengujian. Setelah menyelesaikan tugas-tugas tersebut, responden kemudian mengisi daftar pertanyaan dari SUPR-Qm untuk menilai tingkat usability dari aplikasi yang dikembangkan. Responden dapat memberi nilai pada pertanyaan dengan menggunakan skala *likert*. Adapun hasil pengisian pada SUPR-Qm pada aktor pasien dan konselor masing-masing dijelaskan pada Tabel 6.11 dan Tabel 6.12.

Tabel 6.11 Hasil Pengujian SUPER-Qm Pasien

No	Pertanyaan	Skor					Total Skor
		1	2	3	4	5	
1.	Saya merasa bahwa aplikasi ini penting untuk saya			1	3	1	20
2.	Aplikasi ini merupakan aplikasi pencari konselor psikologi terbaik yang pernah saya gunakan	1		1	2	1	17
3.	Saya tidak tahu apakah ada aplikasi pencari konselor psikologi yang lebih baik dari aplikasi ini	2			3		14
4.	Saya tidak akan menghapus aplikasi ini dari smartphone saya			3	2		17
5.	Saya akan menyarankan aplikasi ini ke teman saya			1		4	23
6.	Saya suka mengeksplorasi fitur-fitur yang terdapat dalam fitur ini			1	2	2	21
7.	Aplikasi ini memiliki seluruh fitur dan fungsi yang saya butuhkan dalam mencari konselor psikologi terdekat		2	2		1	15
8.	Saya akan sering membuka aplikasi ini setiap kali saya ingin		2	1	2		15

	konseling						
9.	Aplikasi ini menyenangkan		2	1	2		15
10.	Aplikasi ini bekerja dengan baik dengan fitur-fitur lain yang ada pada smartphone saya (contohnya gps, peta)			1	1	3	22
11.	Saya akan menggunakan aplikasi ini lagi ketika saya ingin mencari konselor untuk konseling		1		3	1	19
12.	Desain dari aplikasi ini memudahkan saya dalam menemukan informasi yang saya inginkan		2	1	1	1	14
13.	Menurut saya aplikasi ini menarik		1		3	1	19
14.	Kemampuan aplikasi ini sesuai dengan yang saya butuhkan		2	1	2		15
15.	Aplikasi ini sangat mudah dalam hal navigasi		1	2	2		16
16.	Aplikasi ini mudah digunakan		2	1	2		15
Nilai Akhir							277
Nilai Maksimum							400
Nilai SUPER-Qm							69,25%

Tabel 6.12 Hasil Pengujian SUPER-Qm Konselor

No	Pertanyaan	Skor					Total Skor
		1	2	3	4	5	
1.	Saya merasa bahwa aplikasi ini penting untuk saya			1	2	2	21
2.	Aplikasi ini merupakan aplikasi pencari konselor psikologi terbaik yang pernah saya gunakan		1	4			14
3.	Saya tidak tahu apakah ada aplikasi pencari konselor psikologi yang lebih baik dari aplikasi ini		3	1	1		13
4.	Saya tidak akan menghapus aplikasi ini dari smartphone saya			2	1	2	20
5.	Saya akan menyarankan aplikasi				2	3	23

	ini ke teman saya						
6.	Saya suka mengeksplorasi fitur-fitur yang terdapat dalam aplikasi ini			3	2		17
7.	Aplikasi ini memiliki seluruh fitur dan fungsi yang saya butuhkan dalam mencari konselor psikologi terdekat		2	1	2		15
8.	Saya akan sering membuka aplikasi ini setiap kali saya ingin konseling		1	2	1	1	17
9.	Aplikasi ini menyenangkan		1	1	2	1	18
10.	Aplikasi ini bekerja dengan baik dengan fitur-fitur lain yang ada pada smartphone saya (contohnya gps, peta)				2	3	23
11.	Saya akan menggunakan aplikasi ini lagi ketika saya ingin mencari konselor untuk konseling		1	1	1	2	19
12.	Desain dari aplikasi ini memudahkan saya dalam menemukan informasi yang saya inginkan			2	2	1	19
13.	Menurut saya aplikasi ini menarik				4	1	23
14.	Kemampuan aplikasi ini sesuai dengan yang saya butuhkan		2	1	2		15
15.	Aplikasi ini sangat mudah dalam hal navigasi				3	2	22
16.	Aplikasi ini mudah digunakan			2	1	2	20
Nilai Akhir							299
Nilai Maksimum							400
Nilai SUPER-Qm							74,75%

Dari hasil pengujian *usability* yang dilakukan, diperoleh nilai 69,25% untuk aplikasi pada sisi pasien dan 74,75% untuk aplikasi pada sisi konselor. Setelah mengisi pertanyaan yang ada pada SUPR-Qm, responden memberikan kritik dan saran pada aplikasi yang dikembangkan agar dilakukan pengembangan lebih baik

lagi. Adapun kritik dan saran dari responden dijelaskan pada Tabel 6.13 dan Tabel 6.14.

Tabel 6.13 Kritik dan Saran - Pasien

Responden Pasien	Kritik dan Saran
1	Perlu ditampilkan lokasi pertemuan ketika membuat <i>appointment</i> dengan konselor
2	Perlu perbaikan untuk menangani error pada saat mendaftar akun
3	Perlu diperbaiki untuk navigasi pada aplikasi
4	Detail dari <i>user experience</i> perlu ditingkatkan lagi seperti tombol dan <i>wording</i> .
5	Perlu adanya notifikasi untuk mengetahui aktifitas dengan konselor.

Tabel 6.14 Kritik dan Saran - Konselor

Responden Konselor	Kritik dan Saran
1	Fitur konseling online ditambahkan cara komunikasi lewat video call dan telfon
2	Perlu adanya riwayat medis atau riwayat psikologis dari pasien sebagai tambahan informasi bagi konselor dan dikembangkan juga di android
3	Perlu adanya keterangan apakah konselor / pasien sudah membaca pesan atau ketika pengguna sedang mengetik
4	Perlu adanya notifikasi untuk mengetahui aktifitas dengan pasien.
5	Perlu adanya fitur bagi konselor untuk mengatur ketersediaan jadwal <i>appointment</i>

6.2.3 Analisis Hasil Pengujian

Pengujian pada penelitian ini telah selesai dilakukan, kemudian tahap berikutnya adalah melakukan analisis terhadap pengujian yang sudah selesai untuk mendapatkan hasil dan kesimpulannya.

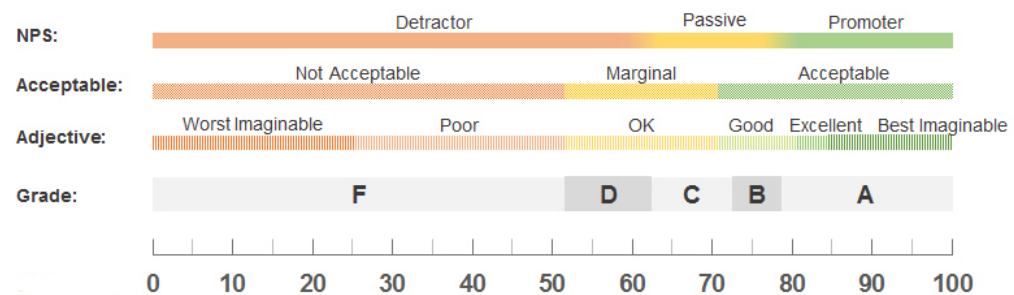
6.2.3.1 Hasil Pengujian *Blackbox*

Pada pengembangan aplikasi yang dilakukan terdapat 4 fitur yang akan dilakukan pengujian *blackbox*. Pengujian yang dilakukan menggunakan *blackbox*

testing dimana peneliti merancang *testcase* dan menguji aplikasi sesuai *testcase* tersebut untuk mengetahui apakah perilaku sistem valid atau tidak. Hasil yang diperoleh adalah 100% valid untuk semua *testcase* dari 4 fitur yang dilakukan pengujian.

6.2.3.2 Hasil Pengujian *Usability*

Setelah dilakukan pengujian *usability*, maka dapat diperoleh nilai atau skor SUPR-Qm. Agar dapat menyimpulkan hasil berdasarkan skor tersebut, maka perlu dilakukan konversi skor tersebut ke dalam *usability rating scale*. *Usability rating scale* dijelaskan pada Gambar 6.1.



Gambar 6.1 Usability Rating Scale

Skor SUPER-Qm dari aplikasi sisi pasien adalah 69,25% maka dapat disimpulkan hasil rating pada aplikasi sisi pasien adalah masuk dalam kategori *ok*. Untuk skor SUPER-Qm dari aplikasi sisi konselor adalah 74,75% maka hasil rating dari aplikasi sisi konselor juga masuk dalam kategori *good*.

BAB 7 PENUTUP

Pada bab penutup menjelaskan hasil atau kesimpulan dari penelitian yang telah dilakukan mulai dari analisis kebutuhan hingga tahap pengujian. Setelah menjelaskan kesimpulan dari penelitian, dilanjutkan menulis saran yang didapatkan agar dapat dilakukan pengembangan lebih lanjut.

7.1 Kesimpulan

Setelah penelitian selesai maka didapatkan hasil kesimpulan yang dimulai dari tahap analisis kebutuhan, perancangan, implementasi, dan pengujian sebagai berikut:

1. Tahap analisis kebutuhan pada penelitian ini menghasilkan 14 kebutuhan fungsional dimana terdiri dari 10 kebutuhan fungsional untuk aplikasi sisi pasien dan 4 kebutuhan fungsional untuk aplikasi sisi konselor. Selain kebutuhan fungsional, terdapat kebutuhan non-fungsional yang didefinisikan pada penelitian ini. Kebutuhan fungsional aplikasi sisi pasien pada penelitian ini terdiri dari proses *login*, *register*, menampilkan peta berisi lokasi pasien dan konselor, menampilkan daftar semua konselor, menampilkan profil konselor, mengirim permintaan konseling, konseling secara online, memberi rating kepada konselor, menampilkan halaman riwayat konseling, dan membuat *appointment* dengan konselor. Kebutuhan fungsional pada aplikasi sisi konselor terdiri dari menampilkan seluruh permintaan konseling, fasilitas untuk menerima atau menolak permintaan konseling, menampilkan profil pasien, dan menampilkan daftar *appointment* dengan pasien. Sedangkan kebutuhan non-fungsional terdiri *usability* yang diukur dengan melalui skema pengujian usabilitas. Kebutuhan tersebut didapatkan dari proses wawancara personal dengan pengguna dan wawancara dengan konselor psikologi.
2. Tahap perancangan sistem pada penelitian ini menghasilkan *class diagram*, *sequence diagram*, rancangan algoritme, rancangan antarmuka sistem, dan rancangan tabel data yang menggunakan struktur data *tree*.
3. Tahap implementasi pada penelitian ini menghasilkan aplikasi perangkat bergerak pada sistem operasi iOS yang memiliki fitur sesuai dengan hasil analisis kebutuhan yang telah dilakukan dengan menggunakan bahasa pemrograman *Swift*.
4. Tahap pengujian pada penelitian ini terdiri dari 2 pengujian yaitu pengujian *blackbox* dan pengujian *usability*. Hasil dari pengujian *blackbox* adalah 100% valid dan hasil pengujian *usability* untuk aplikasi sisi pasien masuk ke dalam kategori *ok* dan aplikasi sisi konselor masuk dalam kategori *good*.

7.2 Saran

Saran yang didapat dari pengembangan aplikasi perangkat bergerak untuk mencari konselor psikologi terdekat adalah sebagai berikut:

1. Implementasi notifikasi untuk memudahkan proses konseling atau interaksi antara pasien dengan konselor.
2. Memberikan informasi yang lebih lengkap pada profil pasien agar memudahkan konselor.
3. Adanya fitur tambahan seperti *video call* atau telfon pada konseling *online*.
4. Perbaiki *user interface* dan *user experience* agar aplikasi lebih menarik dan lebih mudah digunakan.

DAFTAR REFERENSI

- Apple. (n.d.-a). MapKit. Retrieved October 6, 2019, from <https://developer.apple.com/documentation/mapkit>
- Apple. (n.d.-b). Swift The powerful programming language that is also easy to learn. Retrieved October 5, 2019, from <https://developer.apple.com/swift/>
- Apple. (2018). Managing Your App's Life Cycle. Retrieved August 24, 2019, from https://developer.apple.com/documentation/uikit/app_and_environment/managing_your_app_s_life_cycle
- Arjadi, R., Nauta, M. H., & Bockting, C. L. H. (2018). Acceptability of internet-based interventions for depression in Indonesia. *Internet Interventions*, 13(April), 8–15. <https://doi.org/10.1016/j.invent.2018.04.004>
- Arjadi, R., Nauta, M. H., Scholte, W. F., Hollon, S. D., Chowdhary, N., Suryani, A. O., & Bockting, C. L. H. (2016). Guided Act and Feel Indonesia (GAF-ID) - Internet-based behavioral activation intervention for depression in Indonesia: Study protocol for a randomized controlled trial. *Trials*, 17(1), 1–10. <https://doi.org/10.1186/s13063-016-1577-9>
- Avram, A. (2013). iOS vs. Android Development. Retrieved October 5, 2019, from InfoQ website: <http://www.infoq.com/news/2013/08/ios-vs-android-development>
- Bajpai, V., & Gorthi, R. P. (2012). On non-functional requirements: A survey. *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science: Innovation for Humanity, SCEECS 2012*, (April). <https://doi.org/10.1109/SCEECS.2012.6184810>
- Cao, J. (2015). *The Guide to Usability Testing*.
- Chen, M., & Jiang, S. (2019). Analysis and research on mental health of college students based on cognitive computing. *Cognitive Systems Research*, 56, 151–158. <https://doi.org/10.1016/j.cogsys.2019.03.003>
- Erna Kumalasari Nurnawati, J. M. (2014). Aplikasi mobile berbasis lokasi untuk penyedia lokasi layanan kesehatan di Yogyakarta. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)*, 293–300.
- Faisandier, A. (2012). *SYSTEMS OPPORTUNITIES AND REQUIREMENTS*.
- Farouqi, M. I., Aknuranda, I., & Herlambang, A. D. (2018). Evaluasi Usability pada Aplikasi Go-Jek Dengan Menggunakan Metode Pengujian Usability. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(10), 3150–3156. Retrieved from <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/2396/947/>
- Islam, R., & Mazumder, T. (2010). Mobile application and its global impact. *International Journal of Engineering & ...*, (06), 72–78. Retrieved from

<http://ijens.org/107506-0909 IJET-IJENS.pdf>

- Junglas, B. I. A., & Watson, R. T. (2008). *LOCATION-BASED SERVICES*. 51(3), 65–70.
- Larrea, M. (2017). Black-Box Testing Technique for Information Visualization. Sequencing Constraints with Low-Level Interactions. *Journal of Computer Science and Technology*, 17(1), 37–48.
- Mahoney, J., Le Moignan, E., Long, K., Wilson, M., Barnett, J., Vines, J., & Lawson, S. (2019). Feeling alone among 317 million others: Disclosures of loneliness on Twitter. *Computers in Human Behavior*, 98(February), 20–30. <https://doi.org/10.1016/j.chb.2019.03.024>
- Mevada, D. (2018). The Benefits of Having Firebase for Mobile App Development. Retrieved September 6, 2019, from <https://www.mindinventory.com/blog/benefits-of-firebase-in-mobile-app-development/>
- Microsoft. (2017). The Model-View-ViewModel Pattern. Retrieved October 5, 2019, from <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>
- Nielson, J., & Landauer, J. (1993). A mathematical model of finding the usability problem. Proceedings of the CHI 93 proceedings of the Interact conference on human factors in computing systems. *Espacio de Trabajo Matemático. Quinto Simposio Internacional ET*, 206–213. <https://doi.org/10.1145/169059.169166>
- Nugroho, S. C., Nurhayati, O. D., & Widiyanto, E. D. (2017). Aplikasi Pencarian Rute Perguruan Tinggi Berbasis Android Menggunakan Location Based Service (LBS) di Kota Semarang. *Jurnal Teknologi Dan Sistem Komputer*, 3(2), 311. <https://doi.org/10.14710/jtsiskom.3.2.2015.311-319>
- Oinas-Kukkonen, H., & Kurkela, V. (2003). Developing successful mobile applications. *Proceedings of the IASTED International Conference on Computer Science and Technology*, (January 2003), 50–54.
- Rahman. (2015). Android Kuasai Asia Tenggara, di Indonesia Paling Juara.
- Rather, M. A., & Bhatnagar, V. (2016). *A comparative study of sdlc model*. (October 2015).
- Sauro, J., & Zarolia, P. (2017). SUPR-Qm: A Questionnaire to Measure the Mobile App User Experience. *Journal of Usability Studies*, 13(1), 17–37.
- Shivaram, A. M., & Handigund, S. M. (2015). An ameliorated methodology for the abstraction of object oriented features from software requirements specification. *Procedia Computer Science*, 62(Scse), 274–281. <https://doi.org/10.1016/j.procs.2015.08.450>
- Tristiana, R. D., Yusuf, A., Fitryasari, R., Wahyuni, S. D., & Nihayati, H. E. (2018). Perceived barriers on mental health services by the family of patients with

mental illness. *International Journal of Nursing Sciences*, 5(1), 63–67.
<https://doi.org/10.1016/j.ijnss.2017.12.003>

WHO. (2012). *Suicide mortality rate (per 100,000 population)*.

LAMPIRAN A HASIL EXPERT INTERVIEW

Narasumber : Dita Rachmayani, S.Psi., M.A.
Jabatan : Dosen Psikologi UB dan Konselor
Tanggal : 06 Desember 2019
Waktu : 10.30 - 11.00

1. Menurut Anda, apakah aksesibilitas pengobatan kesehatan mental setara dengan aksesibilitas pengobatan kesehatan fisik?
2. Apakah menurut Anda kesehatan mental harus diprioritaskan lebih banyak di Indonesia?
3. Sebagai masyarakat, bagaimana kita bisa meningkatkan kesadaran tentang kepentingan kesehatan mental di dalam negeri?
4. Selama ini apakah ada seseorang yang berkonsultasi atau konseling dengan anda terkait permasalahan kesehatan mentalnya?
5. Apakah penting untuk seseorang yang memiliki masalah pada kesehatan mentalnya untuk bisa bercerita kepada seorang professional?
6. Bagaimana pendapat anda apabila ada sebuah aplikasi mobile yang dapat memudahkan masyarakat dalam mencari psikolog untuk bercerita sekedar untuk pertolongan pertama pada permasalahannya?
7. Apa saja anda butuhkan selaku psikolog pada aplikasi yang bisa membantu masyarakat untuk bercerita tentang masalahnya?

Jawaban

1. Setau saya kalau di Malang tidak semua puskesmas ada psikolognya. Jadi itu menjadi salah satu bukti kalau aksesibilitas untuk kesehatan mental masih kurang.

2. Iya, dan sekarang memang sudah banyak psikolog yang paham tentang isu kesehatan mental dan menurut saya langkah pertamanya adalah memberikan edukasi terkait kesehatan mental di Indonesia melalui sosial media.
3. Dengan semakin bijak menggunakan sosial media karena sekarang marak sekali seperti body shaming dan sebagainya.
4. Ada, namun disini saya sebagai konselor bukan psikolog. Jadi dari mahasiswa UB bahkan mahasiswa psikolog itu sendiri juga banyak yang konseling dengan saya.
5. Iya, menurut saya kalau seseorang mengalami kesulitan seperti itu segera mencari pertolongan ke konselor atau professional.
6. Menurut saya memang penting, akan tetapi ada beberapa kendala ketika klien kesulitan untuk menjelaskan permasalahannya secara rinci. Harapan saya kelengkapan informasi seperti riwayat atau masalah internal klien ada di aplikasinya.
7. Kalau menurut saya yang paling penting adalah kelengkapan informasi klien agar sebagai konselor dapat mencari akar permasalahan dari klien sehingga bisa melakukan konseling secara efektif.

Penulis

Narasumber

Tifo Audi Alif Putra

Dita Rachmayani, S.Psi., M.A.

LAMPIRAN B HASIL PENGUJIAN USABILITAS

Tabel 7.1 Hasil Pengujian Usabilitas Pasien

No	Pertanyaan	Pasien 1	Pasien 2	Pasien 3	Pasien 4	Pasien 5
1.	Saya merasa bahwa aplikasi ini penting untuk saya	5	4	3	4	4
2.	Aplikasi ini merupakan aplikasi pencari konselor psikologi terbaik yang pernah saya gunakan	4	3	1	4	5
3.	Saya tidak tahu apakah ada aplikasi pencari konselor psikologi yang lebih baik dari aplikasi ini	1	4	1	4	4
4.	Saya tidak akan menghapus aplikasi ini dari smartphone saya	3	3	3	4	4
5.	Saya akan menyarankan aplikasi ini ke teman saya	5	5	3	5	5
6.	Saya suka mengeksplorasi fitur-fitur yang terdapat dalam aplikasi ini	5	3	4	4	5
7.	Aplikasi ini memiliki seluruh fitur dan fungsi yang	3	2	2	3	5

	saya butuhkan dalam mencari konselor psikologi terdekat					
8.	Saya akan sering membuka aplikasi ini setiap kali saya ingin konseling	3	2	2	4	4
9.	Aplikasi ini menyenangkan	2	3	2	4	4
10.	Aplikasi ini bekerja dengan baik dengan fitur-fitur lain yang ada pada smartphone saya (contohnya gps, peta)	5	5	4	3	5
11.	Saya akan menggunakan aplikasi ini lagi ketika saya ingin mencari konselor untuk konseling	4	4	2	4	5
12.	Desain dari aplikasi ini memudahkan saya dalam menemukan informasi yang saya inginkan	2	2	3	4	5
13.	Menurut saya aplikasi ini menarik	4	4	2	4	5
14.	Kemampuan aplikasi ini sesuai dengan yang saya	2	3	2	4	4

	butuhkan					
15.	Aplikasi ini sangat mudah dalam hal navigasi	4	2	3	4	3
16	Aplikasi ini mudah digunakan	2	2	3	4	4

Tabel 7.2 Hasil Pengujian Usabilitas Konselor

No	Pertanyaan	Konselor1	Konselor2	Konselor3	Konselor4	Konselor5
1.	Saya merasa bahwa aplikasi ini penting untuk saya	4	3	5	5	4
2.	Aplikasi ini merupakan aplikasi pencari konselor psikologi terbaik yang pernah saya gunakan	2	3	3	3	3
3.	Saya tidak tahu apakah ada aplikasi pencari konselor psikologi yang lebih baik dari aplikasi ini	3	2	2	4	2
4.	Saya tidak akan menghapus aplikasi ini dari smartphone saya	3	5	3	5	4
5.	Saya akan menyarankan aplikasi ini ke teman saya	5	5	5	4	4

6.	Saya suka mengeksplorasi fitur-fitur yang terdapat dalam aplikasi ini	3	4	4	3	3
7.	Aplikasi ini memiliki seluruh fitur dan fungsi yang saya butuhkan dalam mencari konselor psikologi terdekat	4	3	4	2	2
8.	Saya akan sering membuka aplikasi ini setiap kali saya ingin konseling	3	2	4	5	3
9.	Aplikasi ini menyenangkan	2	3	4	5	4
10.	Aplikasi ini bekerja dengan baik dengan fitur-fitur lain yang ada pada smartphone saya (contohnya gps, peta)	5	5	4	5	4
11.	Saya akan menggunakan aplikasi ini lagi ketika saya ingin mencari konselor untuk konseling	5	3	5	4	2
12.	Desain dari aplikasi ini memudahkan saya dalam menemukan	4	4	3	3	5

	informasi yang saya inginkan					
13.	Menurut saya aplikasi ini menarik	4	4	5	4	4
14.	Kemampuan aplikasi ini sesuai dengan yang saya butuhkan	2	3	4	4	2
15.	Aplikasi ini sangat mudah dalam hal navigasi	4	4	4	5	5
16	Aplikasi ini mudah digunakan	3	3	4	5	5