

Desain Antarmuka dan Aplikasi untuk Monitor Lahan Parkir pada Basemen TULT berbasis IoT dan Machine Learning

1ST Pandu Dinan Dwitama

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

pandudwitama@student.telkomuniversi
ty.ac.id2ND Dhoni Putra Setiawan, PH.D., M.T., S.T

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

setiawandhoni@student.telkomuniversity.ac.
id3RD Sri Astuti, S.T., M.T.

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

sriastuti@telkomuniversity.ac.id

Abstrak — Permasalahan utama dalam manajemen parkir di gedung TULT adalah ketidakefisienan dalam mencari tempat parkir yang tersedia, menyebabkan waktu terbuang di area parkir basement. Untuk mengatasi masalah ini, proyek Smart Parking menggunakan teknologi Internet of Things dan Machine Learning, dengan sensor inframerah dan sensor ultrasonik yang terhubung ke Firebase untuk memberikan informasi real-time tentang ketersediaan slot parkir. Data dari sensor ditampilkan melalui layar LCD dan aplikasi mobile, yang juga memprediksi kepadatan parkir menggunakan algoritma Machine Learning. Aplikasi monitoring ini akan memudahkan pengemudi untuk mencari lahan parkir sebelum memasuki basemen TULT. Dengan demikian, pengemudi dapat menghemat waktu untuk mencari parkir.

Kata kunci— Internet of Things, Machine Learning, Firebase, Aplikasi Mobile.

I. PENDAHULUAN

Mencari lahan parkir seringkali menjadi tantangan bagi pengemudi, terutama di gedung Telkom University Landmark Tower (TULT) yang masih menggunakan cara konvensional. Hal ini dapat menghambat mobilitas, khususnya bagi dosen dan orang lain yang memiliki keperluan di gedung tersebut[1].

Pada pelaksanaan Tugas Akhir ini, Penulis mengembangkan sistem Smart Parking dengan berbasis *Internet of Things* (IoT) dan Machine Learning yang nantinya dapat dimonitor melalui aplikasi secara *real-time*. *Internet of Things* (IoT) memungkinkan perangkat saling berkomunikasi dan bertukar informasi secara otomatis dari jarak jauh. Machine learning, bagian dari kecerdasan buatan (AI), memungkinkan komputer belajar dari data untuk membuat prediksi lahan parkir dengan mengumpulkan data di hari sebelum – sebelumnya.

Komponen alat disimpan dalam kotak dan diltekkan di dinding untuk mendeteksi keberadaan mobil yang sedang parkir dan dapat dimonitor melalui aplikasi. Tujuan adanya aplikasi ini agar pengemudi dapat melihat kondisi lahan parkir yang tersedia di basemen sehingga pengemudi tidak

perlu memutari parkiran untuk mencari parkir. Aplikasi dirancang dengan *User Interface* (UI) yang mudah digunakan agar pengemudi dapat menggunakan aplikasi dengan mudah.

II. KAJIAN TEORI

A. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) yang disediakan oleh Google untuk memudahkan pengembangan aplikasi dan didasarkan pada IntelliJ IDEA, sebuah IDE yang dikembangkan oleh JetBrains dengan bahasa pemrograman yang didukung seperti Java, Kotlin, dan C++. Salah satu fitur unggulannya yaitu *desainer layout visual* yang memungkinkan pengembang dapat mendesain antarmuka pengguna aplikasi secara *drag-and-drop*. Android Studio juga sudah terintegrasi dengan Firebase, platform pengembangan aplikasi berbasis *cloud* yang menyediakan berbagai layanan seperti authentication, database, dan analytics sehingga pengembang dapat lebih mudah untuk menambahkan fitur – fitur tersebut ke dalam aplikasi.

B. Figma

Figma merupakan alat desain berbasis *cloud* yang digunakan untuk membuat desain antarmuka pengguna (UI). Karena berbasis *cloud*, Figma dapat diakses melalui browser web tanpa perlu menginstal perangkat lunak khusus sehingga dapat dikerjakan secara bersamaan dalam satu file. Antarmuka pengguna sangat penting karena berinteraksi secara langsung dengan pengguna aplikasi, jika tampilan pengguna dapat dengan mudah memahami fitur – fitur yang ada pada aplikasi maka desain antarmuka dapat dikatakan berhasil menyampaikan informasi melalui fitur – fitur tersebut. Sebuah sistem dengan tampilan antarmuka yang buruk dapat memengaruhi intensitas interaksi pengguna dengan aplikasi karena tampilan yang sulit untuk dipahami sehingga pengguna sulit menggunakan sistem tersebut.

C. Internet of Things (IoT)

Internet of Things adalah suatu konsep dimana koneksi internet dapat bertukar informasi satu sama lainnya dengan benda-benda yang ada di sekelilingnya seperti perangkat sensor ke internet[2]. *Internet of Things* beroperasi dengan memanfaatkan instruksi atau perintah pemrograman dari perangkat sensor yang setiap perintahnya bisa menghasilkan bahasa yang dapat dimengerti ke sesama perangkat terhubung secara otomatis tanpa adanya campur tangan atau ikut campur pengguna, bahkan dalam jarak jauh sekali pun[4]. Prosesnya dimulai dengan pengumpulan data dari sensor kemudian data dikirimkan ke database, lalu database menyimpan dan mengolah data tersebut.

D. Machine Learning

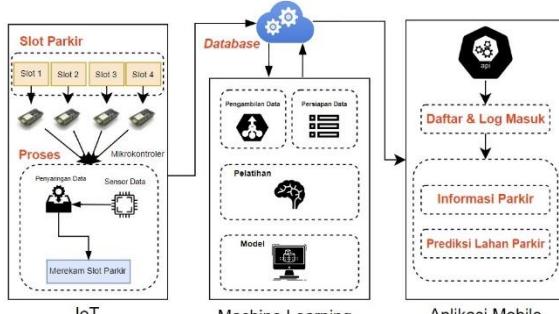
Machine Learning merupakan kecerdasan buatan atau biasa disebut *Artificial Intelligence* (AI) yang memungkinkan sistem untuk belajar dari data tanpa harus diprogram secara eksplisit. Dengan kata lain, algoritma dalam komputer diberikan data dalam jumlah besar dan di *training* untuk menemukan pola dari data tersebut. *Machine Learning* pada sistem ini digunakan untuk membuat prediksi lahan parkir berdasarkan hari – hari sebelumnya dan akan dikembangkan untuk di hari kedepannya atau pekan depannya.

E. User Experience Questionnaire (UEQ)

User Experience Questionnaire (UEQ) adalah sebuah metode evaluasi yang melibatkan serangkaian pertanyaan yang berfokus pada pengalaman pengguna (UX). Metode ini digunakan untuk menilai opini dan pengalaman pengguna terhadap aplikasi dengan mempertimbangkan enam aspek penilaian. Prosesnya dimulai dengan menyusun pertanyaan kuesioner yang mencakup aspek tersebut. Setelah itu, pengguna diminta untuk menilai antarmuka, kemudahan penggunaan, serta kesesuaian aplikasi dengan tujuan yang diharapkan. Data yang diperoleh dari kuesioner kemudian dianalisis untuk mengidentifikasi kelebihan dan kekurangan aplikasi.

III. METODE

A. Diagram Blok Sistem



GAMBAR 1.
DIAGRAM BLOK SISTEM SMART PARKING

Diagram blok di atas menggambarkan arsitektur keseluruhan dari sistem manajemen parkir yang mengintegrasikan teknologi *Internet of Things* (IoT), *Machine Learning*, dan aplikasi *mobile* untuk menampilkan layanan prediksi dan informasi mengenai ketersediaan lahan parkir. Sistem ini terdiri dari tiga komponen utama yang bekerja secara kolaboratif, yaitu:

1. Internet of Things (IoT)

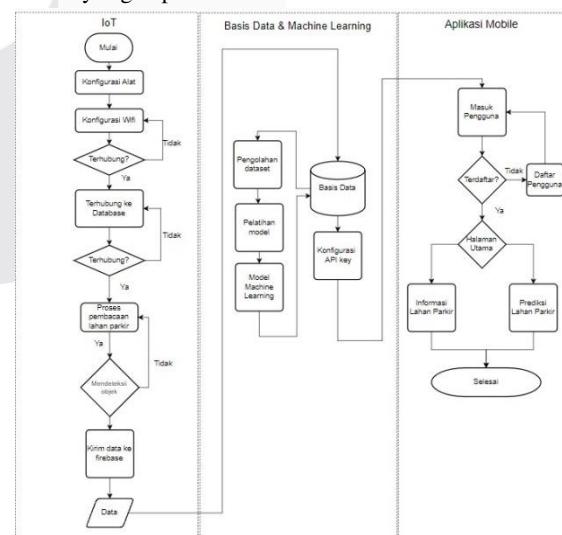
Tahap pertama adalah pengumpulan data yang dilakukan oleh node IoT yang terpasang pada setiap slot parkir. Setiap slot dilengkapi dengan sensor untuk mendeteksi keberadaan kendaraan. Sensor-sensor ini terhubung ke mikrokontroler, yang bertanggung jawab untuk mengumpulkan data dari sensor. Data yang dikumpulkan berupa status ketersediaan slot parkir apakah kosong atau terisi. Setelah data dikumpulkan, mikrokontroler akan mengirimkan data tersebut ke database melalui jaringan internet untuk tahap analisis lebih lanjut.

2. Machine Learning

Pada tahap ini, data yang telah terkumpul di database akan dianalisis menggunakan algoritma *machine learning*. Proses ini meliputi pengolahan data, pelatihan model, dan penerapan model untuk membuat prediksi. Data yang dianalisis tidak hanya berupa status saat ini dari setiap slot parkir, tetapi juga pola penggunaan parkir dari hari – hari sebelumnya . Berdasarkan analisis ini, sistem akan menghasilkan prediksi kapan area parkir, seperti basemen TULT, akan penuh pada hari-hari tertentu dan jam-jam tertentu. Hasil prediksi ini kemudian disimpan kembali ke dalam database yang nantinya ditampilkan pada aplikasi.

3. Aplikasi Mobile

Aplikasi berfungsi sebagai antarmuka pengguna yang memanfaatkan data dari database. Melalui aplikasi ini, pengguna dapat melihat informasi terkini mengenai ketersediaan lahan parkir. Selain itu, aplikasi ini juga memiliki fitur untuk memberikan perkiraan waktu-waktu sibuk atau penuh berdasarkan prediksi yang dihasilkan oleh sistem *machine learning*. Untuk memberikan layanan ini, aplikasi mobile berkomunikasi dengan API yang disediakan oleh database, yang memfasilitasi akses ke data yang diperlukan.



GAMBAR 2.
FLOWCHART SISTEM SMART PARKING

Pada gambar diatas menjelaskan tentang alur sistem dari smart parking yang akan diterapkan. Pada awalnya perangkat

IoT akan mempersiapkan alat-alat yang akan digunakan. Setelah itu melakukan konfigurasi untuk menghubungkan jaringan Wifi dan Firebase agar dapat mengirimkan data yang telah diterima oleh alat-alat yang digunakan ke dalam *database*. Setelah data dikumpulkan oleh perangkat IoT, data tersebut akan dikirim ke dalam *database* untuk diolah dengan algoritma *machine learning* agar menghasilkan model prediksi yang akurat. Setelah itu, model yang dihasilkan akan dikirim kembali ke dalam *database* untuk disimpan dan diimplementasikan pada aplikasi seluler. Pada aplikasi seluler akan menampilkan informasi kondisi lahan parkir dan menampilkan fitur prediksi kondisi lahan parkir pada hari-hari tertentu.

B. User Interface (UI)

Pada pengembangan desain antarmuka (UI) aplikasi dibuat menggunakan Figma, sebuah alat desain berbasis cloud yang digunakan untuk merancang desain UI/UX pada aplikasi. Setelah melakukan perancangan tahap selanjutnya yaitu menerapkannya pada Android Studio. Berikut hasil rancangan UI yang telah dibuat.



GAMBAR 3.
DESAIN UI APLIKASI

C. Back End

a. Firebase

Firebase digunakan sebagai pengintegrasian sensor-sensor dan aplikasi untuk mengakses data yang telah dikumpulkan oleh sensor secara realtime.



GAMBAR 4.
REALTIME DATABASE

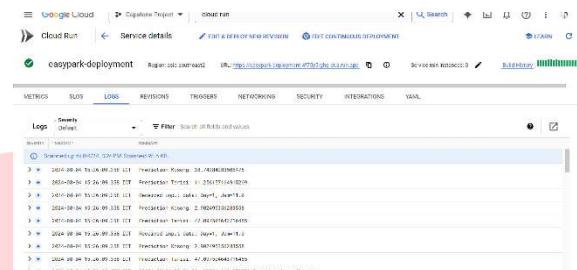
Pada gambar 4 "UltrasonicDistance", "status", dan "timestamp" merupakan elemen data atau disebut *field* yang merupakan bagian dari suatu dokumen agar mikrokontroler mengetahui nilai yang dibaca oleh sensor dengan membaca *value* dari *field* tersebut. *Value* dari *field* "UltrasonicDistance" akan berubah setiap sensor memperoleh data dengan nilai sesuai dengan data yang sensor ambil dan *value* dari *field* "status" akan berubah dari *false* menjadi *true* saat keadaan slot parkir sedang kosong.

Field timestamp pada Firebase akan menampilkan waktu yang dibaca pada mikrokontroler setiap

pengambilan data dari sensor. Data dari *field* ini kemudian diambil oleh aplikasi untuk memonitoring lahan parkir.

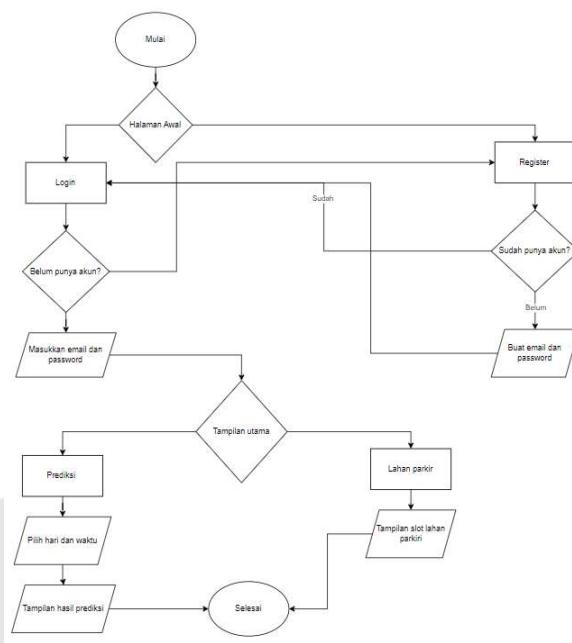
b. Google Cloud Platform (GCP)

Google Cloud Platform digunakan untuk menyimpan model dari Machine Learning yang sudah dulu diolah dalam format TensorFlow yang nantinya akan dimunculkan pada fitur prediksi aplikasi.



GAMBAR 5.
LOGS GOOGLE CLOUD PLATFORM

D. Alur Sistem Aplikasi



GAMBAR 6.
FLOWCHART APLIKASI

Gambar 3 merupakan flowchart yang menjelaskan alur kerja pengoperasian aplikasi. Pengoperasian dimulai dengan memulai aplikasi, setelah itu, halaman awal aplikasi akan muncul dengan pilihan *login* atau *register*, jika pengguna sudah mempunyai akun maka bisa melakukan *login*, jika belum pengguna dapat membuat akun di halaman *register*. Setelah melakukan *login* halaman akan berpindah ke halaman utama dan pengguna dapat memilih untuk melihat kondisi lahan parkir atau prediksi dari lahan parkir dengan mengklik salah satu tombol yang ada di halaman utama yaitu Lahan Parkir dan Prediksi. Jika pengguna mengklik Lahan Parkir tampilan yang muncul yaitu kondisi lahan parkir yang ditandai dengan tanda warna merah dan hijau, jika slot

berwarna merah maka slot tersebut sedang terisi oleh mobil, jika slot berwarna hijau maka slot tersebut sedang kosong. Jika pengguna mengklik Prediksi maka tampilan akan memunculkan prediksi lahan parkir dengan memilih hari dan jam yang tersedia di tampilan Prediksi. Setelah itu, klik Check Parking Status untuk memunculkan hasil prediksi tersebut.

IV. HASIL DAN PEMBAHASAN

Sistem smart parking berbasis Internet of Things (IoT) dan Machine Learning bertujuan agar pengemudi tidak perlu parkir dengan memutari basemen gedung TULT karena dapat memakan waktu yang cukup lama. Dengan adanya sistem ini pengemudi menjadi lebih mudah untuk menentukan parkir di basemen dengan melihat kondisi lahan parkir terlebih dahulu melalui aplikasi, setelah itu, pengemudi bisa langsung menuju basemen untuk parkir jika informasi slot pada aplikasi kosong. Pada sistem ini melibatkan integrasi antara perangkat keras dengan perangkat lunak. Perangkat keras seperti mikrokontroler, sensor jarak, dan sensor inframerah sedangkan perangkat lunak yang digunakan yaitu Firebase Google Cloud Platform (GCP), dan aplikasi untuk monitoring kondisi lahan parkir. Rangkaian prosedur penelitian yang telah dilakukan dan hasil penelitiannya dapat dijelaskan sebagai berikut.

A. Pengujian Fungsionalitas Sistem Aplikasi

1. Fitur *Login*

Pengujian fitur *login* berfungsi untuk memastikan apakah fitur ini berjalan dengan baik atau tidak. Indikator keberhasilan fitur ini yaitu dengan pengguna dapat masuk melalui akun yang terdaftar saat pengguna membuat akun.

```
binding.btnLogin.setOnClickListener {
    val email = binding.emailLogin.text.toString()
    val password = binding.passwordLogin.text.toString()

    if (email.isEmpty()) {
        binding.emailLogin.error = "Email Harus Terisi"
        binding.emailLogin.requestFocus()
        return@registerOnClickListener
    }
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        binding.emailLogin.error = "Email Tidak Valid"
        binding.emailLogin.requestFocus()
        return@registerOnClickListener
    }
    if (password.isEmpty()) {
        binding.passwordLogin.error = "Password Harus Terisi"
        binding.passwordLogin.requestFocus()
        return@registerOnClickListener
    }

    LoginFirebase(email, password)
}
```

GAMBAR 9.

FITUR *LOGIN* APLIKASI

Pada Gambar 9 merupakan kode dari fitur *login* untuk pengguna dengan mengisi *email* dan *password* lalu klik tombol *Login*. Tombol *Login* ini akan mengirimkan *email* dan *password* ke *Firebase* sehingga jika *email* dan *password* sesuai dengan daftar nama yang telah buat akun maka pengguna dapat login untuk memasuki halaman utama

2. Fitur *Register*

Pengujian fitur *register* berfungsi untuk memastikan apakah pengguna bisa mendaftarkan akunnya atau tidak dengan mendaftar sebagai pengguna baru dan mengamati pada *Authentication* di *Firebase* bagian *user*. Jika ada daftar nama pengguna maka akun pengguna sudah terdaftar.

```
binding.btnConfirm.setOnClickListener {
    val email = binding.email.text.toString()
    val password = binding.password.text.toString()

    if (email.isEmpty()) {
        binding.email.error = "Email Harus Terisi"
        binding.email.requestFocus()
        return@registerOnClickListener
    }
    if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        binding.email.error = "Email Tidak Valid"
        binding.email.requestFocus()
        return@registerOnClickListener
    }
    if (password.isEmpty()) {
        binding.password.error = "Password Harus Terisi"
        binding.password.requestFocus()
        return@registerOnClickListener
    }
    if (password.length < 5) {
        binding.password.error = "Minimal 5 Karakter"
        binding.password.requestFocus()
        return@registerOnClickListener
    }

    RegisterFirebase(email, password)
}
```

GAMBAR 10.
FITUR *REGISTER* APLIKASI

Pada Gambar 9 adalah code untuk registrasi akun pengguna baru dan mengklik tombol *Confirm*. Logika pada tombol *Confirm* sama dengan tombol *Login* yaitu setelah pengguna registrasi akun dengan mengisi *email* dan *password* menekan tombol *Confirm* dan tombol ini akan mengirimkan data tersebut ke *Firebase* untuk disimpan pada *Authentication* *Firebase* bagian *user*.

3. Fitur *Monitoring*

Fitur ini berfungsi untuk menampilkan nilai hasil dari bacaan sensor. Pengujian ini dilakukan untuk memastikan nilai sensor pada *Realtime Database* *Firebase* dapat muncul pada fitur ini sesuai dengan bacaan sensor.

```
private fun loaddatafromrealtime() {
    parkingSlotRef.addValueEventListener(object : ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            val status1 =
                snapshot.child("slot1").child("status").getValue(Boolean::class.java) ?: false
            val status2 =
                snapshot.child("slot2").child("status").getValue(Boolean::class.java) ?: false
            val status3 =
                snapshot.child("slot3").child("status").getValue(Boolean::class.java) ?: false
            val status4 =
                snapshot.child("slot4").child("status").getValue(Boolean::class.java) ?: false

            val time1 = snapshot.child("slot1").child("timestamp").value.toString()
            val time2 = snapshot.child("slot2").child("timestamp").value.toString()
            val time3 = snapshot.child("slot3").child("timestamp").value.toString()
            val time4 = snapshot.child("slot4").child("timestamp").value.toString()

            binding.apply {
                slot1Image = if (status1) R.drawable.lahan_kosong else R.drawable.lahan_kosong
                slot2Image = if (status2) R.drawable.lahan_kosong else R.drawable.lahan_kosong
                slot3Image = if (status3) R.drawable.lahan_kosong else R.drawable.lahan_kosong
                slot4Image = if (status4) R.drawable.lahan_kosong else R.drawable.lahan_kosong
            }
        }
    })
}
```

GAMBAR 11.
FITUR *MONITORING*

Pada Gambar 11 merupakan code untuk melihat kondisi lahan parkir yang ditandai dengan "val status1", "val status2", "val status3", dan "val status 4". Untuk mempermudah membedakan slot kosong dan terisi penulis menggunakan indikator hijau untuk kosong dan merah untuk terisi dengan code "val slot1Image", "val slot2Image", "val slot3Image", dan "val slot4Image". Val time pada code tersebut digunakan untuk menampilkan waktu dari setiap sensor mengambil data dalam format waktu "Jam : Menit : Detik". Pengujian ini dilakukan dengan mengamati perubahan dari indikator val slotImage dengan *Realtime Database* *Firebase* jika value dari *field* status di *Realtime Database* *false* maka val slotImage akan menampilkan indikator berwarna hijau dan sebaliknya jika value dari *field* status *true* maka val slotImage akan menampilkan indikator merah.

4. Fitur *Prediksi*

Fitur prediksi berfungsi untuk menampilkan hasil prediksi dari model *machine learning* yang telah

ditraining. Indikator keberhasilan dari pengujian adalah hasil prediksi dapat muncul pada fitur prediksi.

```

private lateinit var predictionRepository: PredictionRepository

override fun onCreateView(
    inflater: LayoutInflater, container: ViewGroup?,
    savedInstanceState: Bundle?
): View? {
    return inflater.inflate(R.layout.fragment_prediksi, container, false)
}

override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)

    val spinnerDay: Spinner = view.findViewById(R.id.spinner_day)
    val spinnerTime: Spinner = view.findViewById(R.id.spinner_time)
    val buttonPredict: Button = view.findViewById(R.id.button_submit)
    val textViewResult: TextView = view.findViewById(R.id.text_view_result)

    predictionRepository = PredictionRepository()

    buttonPredict.setOnClickListener { view: View? ->
        val selectedDayNumeric = convertDayToNumeric(spinnerDay.selectedItem.toString())
        val selectedTime = spinnerTime.selectedItem.toString()
        val formattedTime = utils.formatTimeForAPI(selectedTime)
        fetchPrediction(selectedDayNumeric, formattedTime, textViewResult)
    }
}

```

GAMBAR 12.
FITUR MONITORING

Di dalam PredictionRepository, prediksi dari lahan parkir ditampilkan berdasarkan pemilihan dari “val spinnerDay” dan “val spinnerTime” yang merupakan hari dan jam lalu mengklik tombol Check Parking Status untuk menampilkan hasil dari prediksi berdasarkan pemilihan tersebut. Pengujian ini memastikan apakah hasil prediksi dapat muncul atau tidak dengan melakukan metode tersebut.

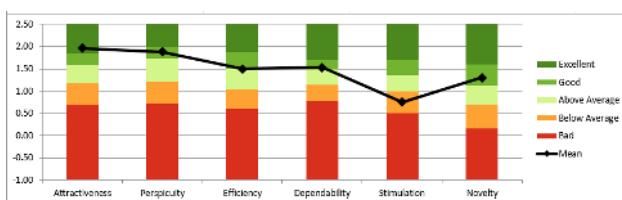
B. Pengujian User Experience Questionnaire (UEQ)

Pengujian User Experience Questionnaire (UEQ) merupakan sebuah metode pengujian dengan menyajikan pertanyaan-pertanyaan mengenai *User Experience* (UX) dan ada enam aspek penilaian di dalamnya[3]. Pengujian dilakukan dengan menyajikan kuisioner. Terdapat 30 responden kemudian jawaban tersebut diolah menggunakan alat analisis UEQ pada Microsoft Excel untuk mendapatkan nilai rata – rata sebagai analisis pengujian tersebut. Aspek penilaian dari kuisioner tersebut seperti daya tarik, kejelasan, efisiensi, ketepatan, stimulasi, dan keterbaruan.

Skala UEQ (Mean and Variance)		
Daya Tarik	1.967	0.65
Kejelasan	1.883	1.65
Efisiensi	1.500	2.81
Ketepatan	1.533	1.98
Stimulasi	0.756	0.10
Keterbaruan	1.300	6.06

GAMBAR 13.
RATA – RATA HASIL SKALA UEQ

Gambar 13 adalah rata – rata dari skala nilai hasil UEQ. Kesimpulan yang didapatkan dari rata – rata tersebut nilai daya tarik sebesar 1.967, kejelasan sebesar 1.883, efisiensi sebesar 1.500, ketepatan sebesar 1.533, stimulasi sebesar 0.756, dan keterbaruan sebesar 1.300. Nilai tersebut diasumsikan memiliki nilai maksimal 3.5.



GAMBAR 14.
GRAFIK SKALA UEQ

Gambar 14 merupakan hasil rata – rata skala UEQ jika dibuatkan menjadi grafik. Dari grafik tersebut dapat disimpulkan Skala daya tarik mendapatkan nilai rata-rata tertinggi, menunjukkan bahwa produk ini sangat menarik bagi sebagian besar pengguna. Skala kejelasan juga mendapat penilaian positif, menunjukkan bahwa aplikasi ini mudah digunakan. Skala efisiensi dan ketepatan menunjukkan bahwa aplikasi ini dapat diandalkan. Pada aspek stimulasi mendapatkan nilai yang kurang dan perlu adanya perbaikan dan konsistensi dalam inovasi untuk memberikan pengalaman pengguna yang lebih memuaskan.

V. KESIMPULAN

Aplikasi pada sistem smart parking berfungsi untuk memonitor lahan parkir secara realtime pada basemen gedung TULT dan dapat memprediksi kondisi lahan parkir berdasarkan hari – hari sebelumnya dengan algoritma machine learning.

Hasil pengujian menunjukkan bahwa aplikasi yang terhubung dengan Firebase dan machine learning berfungsi dengan baik secara keseluruhan. Fitur autentikasi pengguna menggunakan Firebase Authentication berjalan mulus tanpa kendala, sementara proses penyimpanan dan pengambilan data dari Firebase Realtime Database berlangsung dengan cepat dan real-time. Secara umum, aplikasi ini telah memenuhi harapan fungsionalitas yang diinginkan, meskipun masih ada ruang untuk peningkatan lebih lanjut dalam hal kinerja dan pengalaman pengguna. Untuk pengembangan selanjutnya dapat meningkatkan aspek stimulasi pada aplikasi agar memberikan pengalaman pengguna yang lebih memuaskan. Selain itu, tampilan pada aplikasi juga dapat dikembangkan agar tampilan antarmuka bisa lebih menarik dengan fitur – fitur baru yang lebih modern.

Dengan dilakukannya penelitian ini diharapkan dapat mempermudah pengguna lahan parkir basemen TULT untuk mencari lahan parkir lebih cepat. Selain memberikan hasil yang positif, tetapi juga memberikan kesan inovatif untuk pengembangan selanjutnya.

REFERENSI

- [1] D. Hernikawati, “Perbandingan Solusi Parkir Konvensional Dengan Smart Parking,” Open Journal Systems, pp. 118–130, Dec. 2021.
- [2] Yudho Yudha, A. Azis, and E. H. Pratisto, “Pengantar Teknologi Internet Of Things (IoT),” UNS Press. Accessed: Oct. 18, 2023. [Online]. Available: <https://d3tisolo.vokasi.uns.ac.id/pr/product/pengantar-teknologi-internet-of-things-iot>
- [3] Edi Susilo, “Cara Menggunakan User Experience Questionnaire (UEQ) Pada Uji UX.” Accessed: Jul. 31, 2024. [Online]. Available: <https://www.edisusilo.com/cara-menggunakan-user-experience-questionnaire/>