

Aplikasi Monitoring Data Cuaca Pada Smart Weather Station Menggunakan MIT App Inventor

1st Nunung Puspa Sari
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

nununghermawan@student.telkomuniversity.ac.id

2nd Nachwan Mufti Adriansyah
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

nahcwanma@telkomuniversity.ac.id

3rd Vinsensius Sigit Widhi Prabowo
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

vinsensiusvsw@telkomuniversity.ac.id

Abstrak — Dalam penelitian ini, melakukan pengembangan dan pengujian aplikasi Skycast yang dapat memonitoring cuaca secara *realtime* dengan menggunakan perangkat IoT berbasis ESP32 yang dilengkapi dengan berbagai sensor seperti sensor BMP180 untuk tekanan udara, sensor DHT11 untuk suhu dan kelembaban serta sensor *raindrop* untuk curah hujan. Data yang didapatkan disimpan dan diakses pada *Firestore Realtime Database*. Aplikasi memudahkan pengguna untuk mengakses data cuaca secara langsung menggunakan *android*, pengujian yang dilakukan menggunakan beberapa scenario untuk memastikan fungsional aplikasi, kecepatan respon terhadap perubahan data pada sensor, serta kemampuan aplikasi dalam menampilkan data-data histori dan data *realtime* dengan akurasi tinggi yang menunjukkan bahwa aplikasi ini efektif dan memenuhi kebutuhan pengguna sesuai dengan Tingkat Efektivitas yang diperoleh yaitu 71%. Dimana untuk pencapaiannya dinilai efektif yang diukur menggunakan metode *System Usability Scale (SUS)*. Meskipun terdapat beberapa aspek yang masih bisa ditingkatkan untuk memperbaiki pengalaman pengguna dalam menggunakan aplikasi serta meningkatkan skor SUS di penelitian berikutnya.

Kata kunci— *Iot, Realtime, System Usability Scale (SUS)*

I. PENDAHULUAN

Mengetahui kondisi cuaca terkini menjadi sangat penting bagi banyak orang, karena berbagai aktivitas sehari-hari sering kali bergantung pada cuaca di sekitar. Perubahan cuaca ekstrem yang cepat membuat kebutuhan akan informasi cuaca menjadi hal yang sangat dibutuhkan. Laporan cuaca yang perlu diketahui setiap hari mencakup suhu, kelembaban, curah hujan, dan tekanan udara. Pengamatan ini digunakan sebagai dasar untuk memprediksi kondisi cuaca di masa depan.

Dalam era *digital* saat ini, informasi apa pun dapat diakses dengan mudah dan cepat melalui *internet*, termasuk dari perangkat genggam seperti *smartphone* berbasis *Android*. Dengan *smartphone*, pengguna dapat mengakses berbagai informasi secara luas dalam waktu singkat.

Dengan latar belakang ini, dikembangkanlah aplikasi *Android* untuk sistem pemantauan cuaca yang memanfaatkan

Smart Weather Station berbasis ESP32 dan konektivitas Wi-Fi. Sistem ini menggunakan sensor suhu, kelembaban, curah hujan, dan tekanan udara untuk mengumpulkan data cuaca yang kemudian dikirim ke *cloud* untuk pemantauan *real-time*. Pengguna dapat mengakses informasi cuaca secara *fleksibel* melalui aplikasi *Android* atau *website* kapan saja dan di mana saja, dengan dukungan koneksi *internet* yang stabil.

II. KAJIAN TEORI

A. Monitoring

Monitoring adalah kegiatan yang melibatkan observasi, pemantauan, dan pengumpulan data secara berkala atau konsisten pada suatu kondisi, kegiatan ataupun sistem tertentu. Tujuan utama kegiatan ini adalah untuk memperoleh pemahaman yang lebih mendalam, mengukur serta mengawasi perkembangan yang terjadi dalam situasi atau sistem tertentu. Secara umum, monitoring bertujuan mengumpulkan informasi yang relevan yang terkait dengan parameter atau kondisi tertentu. Tujuan ini dapat bervariasi, seperti memastikan keamanan dalam lingkungan industri, menilai kualitas air di lingkungan, atau memantau tingkat kelembaban dalam proses produksi, dan lain-lain. Dalam beberapa kasus, monitoring perlu dilakukan secara kontinu untuk mengamati perubahan secara *real-time*, sementara dalam situasi lain, pengumpulan data secara berkala sudah cukup.

Pengumpulan data merupakan tahap penting dalam monitoring yang melibatkan berbagai sumber, seperti penerima manfaat, pemangku kepentingan, dan mitra kerja organisasi. Data yang dikumpulkan mencakup ketepatan waktu, substansi program, kualitas pelaksana, serta pemanfaatan program, dan disesuaikan dengan indikator dan target yang telah ditetapkan. Sistem monitoring terintegrasi dalam perencanaan organisasi perlu melibatkan seluruh pihak berkepentingan, dengan dukungan unit monitoring dan evaluasi yang menyediakan instrumen pengumpulan data. Penting untuk memilih sumber verifikasi dan metode pengumpulan data yang sesuai dengan indikator yang dimonitor, mempertimbangkan biaya dan kompleksitasnya. Metode pengumpulan data meliputi berbagai teknik, seperti

review dokumen, survei, wawancara, observasi, dan pemetaan. Sebelum pengumpulan data, disarankan melakukan analisis situasi untuk memahami dinamika penerima manfaat dan kondisi sosial, ekonomi, budaya, serta politik yang relevan.[1]

B. Aplikasi

Aplikasi adalah program yang telah dirancang dan siap digunakan untuk menjalankan fungsi tertentu oleh pengguna. Aplikasi dapat dimanfaatkan oleh berbagai pihak sesuai dengan tujuan penggunaannya. Menurut definisi yang terdapat dalam Kamus Komputer Eksekutif, aplikasi diartikan sebagai solusi untuk menyelesaikan masalah melalui teknik pemrosesan data, yang biasanya berhubungan dengan komputasi, dengan tujuan untuk mencapai hasil yang diinginkan melalui proses pengolahan data.

Menurut KBBI, aplikasi dapat diartikan sebagai implementasi dari suatu rancangan mekanisme yang digunakan untuk memproses dan mengolah data sesuai dengan regulasi atau prosedur dari bahasa pengkodean khusus. Secara umum, aplikasi merupakan perangkat lunak yang dirancang untuk menjalankan fungsi tertentu bagi pengguna.[2]

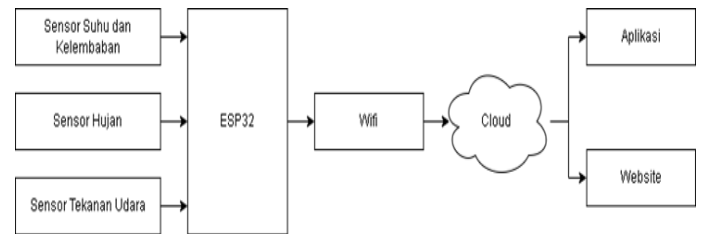
C. MIT App Inventor

MIT App Inventor adalah sebuah platform yang dibuat untuk memudahkan pembuatan aplikasi sederhana tanpa memerlukan pemahaman mendalam tentang bahasa pemrograman. Platform ini berbasis web, memungkinkan pengguna untuk mengembangkan aplikasi tanpa harus menguasai banyak kode. Dengan App Inventor, pengguna dapat merancang aplikasi sesuai kebutuhan dengan menggunakan berbagai pola dan elemen yang tersedia. Awalnya dihentikan oleh Google, namun kemudian diluncurkan kembali sebagai proyek open-source yang kini dikelola oleh Massachusetts Institute of Technology (MIT).

App Inventor memudahkan pengguna untuk melakukan pemrograman dengan tujuan membuat aplikasi perangkat lunak yang berjalan di sistem operasi Android. Platform ini menggunakan pemrograman berbasis blok visual, memungkinkan pengguna untuk merancang dan menyusun blok-blok berisi perintah serta fungsi event handler guna membangun aplikasi yang kompatibel dengan Android. App Inventor sangat membantu bagi pemula dalam dunia pemrograman untuk membuat aplikasi Android. Dengan antarmuka grafis serupa dengan Scratch dan StarLogo TNG, pengguna dapat memindahkan dan menyusun objek visual guna merancang aplikasi yang dapat dijalankan di perangkat Android. Google telah melakukan riset dalam bidang komputasi edukasional dan menyempurnakan lingkungan pengembangan online sebagai bagian dari inovasi App Inventor ini.[3]

III. METODE

Pada riset ini menggunakan sistem penelitian dan pengembangan. Penelitian ini bertujuan untuk mendeteksi cuaca secara langsung dan dapat dimonitoring dimana saja dengan alat yang dipasang pada tower sites. Berikut ada gambaran untuk diagram blok untuk kerja sistem secara keseluruhan.



GAMBAR 1. Global Sistem Keseluruhan

Sistem ini akan mengambil data selama proses deteksi cuaca berlangsung. Sensor-sensor dihubungkan dengan mikrokontroler ESP32 untuk melakukan pengambilan data dari setiap sensor. Hasil dari data tiap sensor akan dikirimkan ke mikrokontroler lalu ke cloud melalui konektivitas internet atau Wifi.

Sistem Smart Weather Station akan bekerja seperti pada gambar 1. Setelah instalasi sensor dan variabel-variabel yang dibutuhkan, sensor akan mengambil data sesuai dengan sensornya. Setelah data didapatkan, data akan dikirim ke cloud. Pada sistem cloud akan mengirim data ke website dan aplikasi yang diintegrasikan dengan jaringan internet atau wifi.

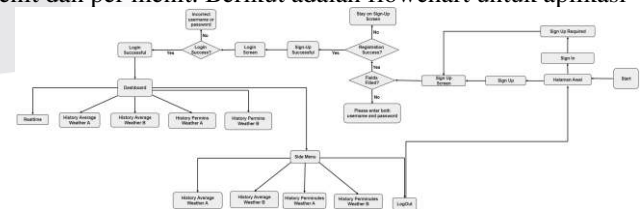
Adapun penjelasan untuk aplikasi akan dijelaskan dibawah terkait kegunaan, cara kerja sistem dan implemetasinya.

Aplikasi Skycast berfungsi untuk menyediakan informasi cuaca secara realtime kepada pengguna. Aplikasi ini memanfaatkan data dari berbagai sensor yang dipasang dilapangan, yang kemudian disimpan dan diakses melalui Firebase Realtime Database[10 pts].

A. Cara Kerja Sistem

Sistem monitoring cuaca ini beroperasi dengan mengintegrasikan beberapa komponen utama, yakni sensor-sensor yang dipasang pada antena atau menara BTS untuk memantau kondisi cuaca secara realtime. sensor-sensor tersebut meliputi sensor suhu dan kelembaban, curah hujan, dan tekanan udara.

Data yang diambil dari firebase sebagai cloud dalam sistem smart weather station dikirim secara nirkabel ke perangkat android yang menjalankan aplikasi untuk memantau kondisi cuaca di sekitar tower sites. Aplikasi ini menerima data, lalu menampilkan secara menyeluruh pada screen realtime dan untuk historinya terbagi dua yaitu per 30 menit dan per menit. Berikut adalah flowchart untuk aplikasi

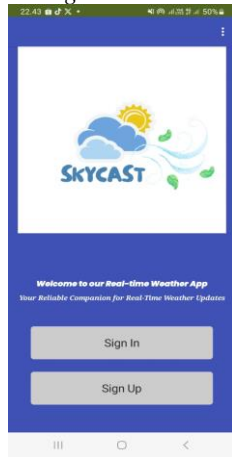


GAMBAR 2. Flowchart Aplikasi

B. Implementasi

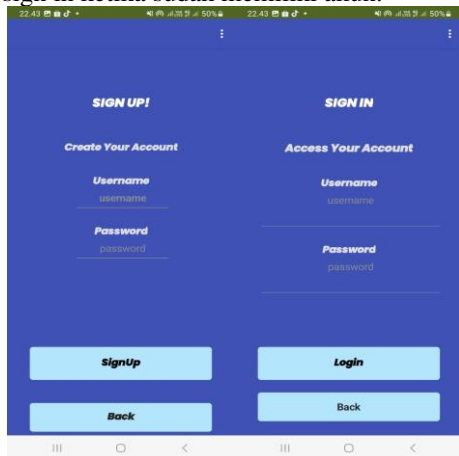
Aplikasi android untuk Skycast ini dibuat menggunakan platform App Inventor (MIT App Inventor). Pada aplikasi ini terdapat main page, sign up, login, main dashboard, realtime history average weather A, History Average Weather B, History Per Minutes Weather A, History Per Minutes

Weather B dan Side menu yang isinya mencari histori-histori yang sudah berlalu serta *logout*.



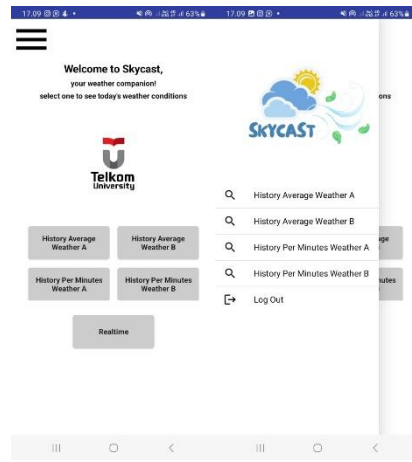
GAMBAR 3.
Main page

Pada menu halaman awal terdapat *button sign in* dan *sign up* terdapat menu untuk membuat akun jika belum memiliki akun, dan *sign in* ketika sudah memiliki akun.



GAMBAR 4.
Menu Sign Up dan Login

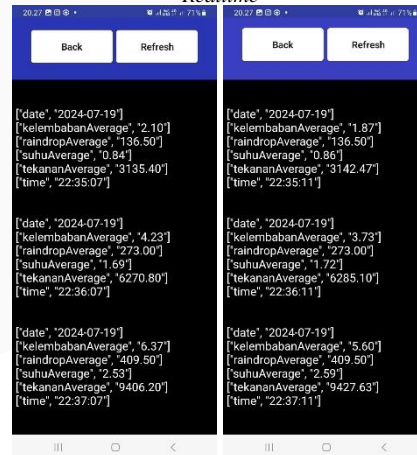
Untuk mengakses fitur-fitur aplikasi pengguna harus terlebih dahulu menyelesaikan proses *signup* dan *signin*. Pada bagian pertama, pengguna memilih tombol "*sign up*" untuk melakukan registrasi. Kemudian, pengguna akan diarahkan untuk mengisi kolom *username* dan *password* sebelum memilih tombol "*sign up*" untuk menyelesaikan proses registrasi. Setelah berhasil membuat akun, pengguna akan dibawa ke tampilan *login*. Di tampilan ini, pengguna mengisi *username* dan *password* yang sudah terverifikasi. Sebelum memilih tombol "*Login*" untuk membuka aplikasi. Setelah *login* berhasil, pengguna akan diarahkan ke layar *main dashboard*.



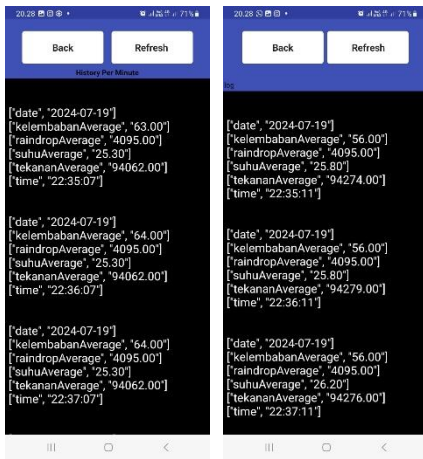
GAMBAR 5.
Main Dashboard



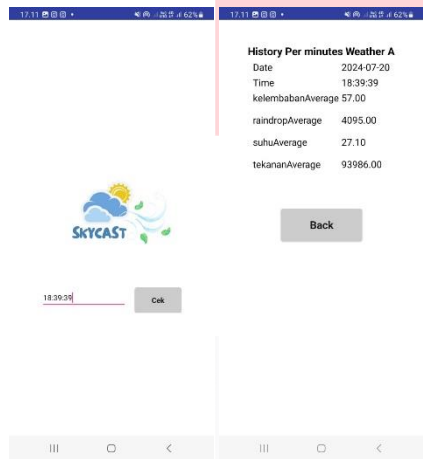
GAMBAR 6.
Realtime



GAMBAR 7.
History Average Weather A dan Weather B



GAMBAR 8. History Per Minutes Weather A dan Weather B



GAMBAR 9. Pencarian History-History

Pada tampilan *main screen*, terdapat 5 tombol yaitu tombol *History Average Weather A*, *History Average Weather B*, *History Per Minutes Weather A*, *History Per Minutes Weather B*, dan *Realtime* dan juga terdapat *side menu* yang menampilkan *navigation drawer* yang memiliki opsi *History Average Weather A*, *History Average Weather B*, *History Per Minutes Weather A*, *History Per Minutes Weather B* untuk mencari data cuaca yang ada di *firebase* pada hari-hari yang sudah lalu dan tombol *logout*, jika ditekan maka akan berpindah ke halaman awal. Dan jika menekan salah satu tombolnya maka akan langsung ke *screen* sesuai dengan nama keterangan tombol yang telah ditekan.

Jika menekan tombol *realtime*, maka pada *screen* akan menampilkan data kelembaban, *raindrop*, suhu dan tekanan yang ada pada *weather A* dan *weather B* secara *realtime* sesuai dengan yang ada pada *firebase*.

Lalu, jika menekan tombol *History Average Weather A* atau *Weather B*, maka akan menampilkan data yang mencakup *date*, *kelembabanAverage*, *raindropAverage*, *suhuAverage*, *tekananAverage* dan *time*. Data ini diambil setiap 30 menit sekali untuk setiap data dari *weather A* dan *weather B*.

Dan, jika menekan tombol *History Per Minutes Weather A* dan *Weather B*, maka akan menampilkan data yang mencakup *date*, *kelembabanAverage*, *raindropAverage*, *suhuAverage*, *tekananAverage* dan *time*. Data ini diambil

setiap satu menit sekali untuk setiap data dari *weather A* dan *weather B*.

IV. HASIL DAN PEMBAHASAN

Pengujian dalam Aplikasi Android bertujuan untuk mengoptimalkan system *smart weather station*. Aplikasi ini memanfaatkan *Firestore Realtime Database* untuk menyimpan data yang diterima dari sensor suhu, kelembaban, *raindrop* dan tekanan secara *real-time* dan *firebase authentication* untuk *setting* pengirim data.

A. Proses Pengujian

1. Pengujian Fungsionalitas Aplikasi, Memastikan bahwa semua fitur pada aplikasi telah dikembangkan dan berfungsi dengan baik.
2. Memastikan Pengambilan Data dari *Firestore Realtime Database*, Memastikan bahwa data yang diambil dari *Firestore Realtime Database* ditampilkan dengan benar dalam aplikasi dan *firebase authentication* untuk *setting* pengirim data.
3. Memverifikasi Koneksi dan Integrasi Aplikasi dengan *Firestore Realtime Database*, bahwa aplikasi terintegrasi dengan *Firestore Realtime Database* untuk penyimpanan dan pengambilan data sensor.
4. Pengujian Kecepatan dan Respon Aplikasi terhadap Perubahan Data Sensor serta Pengambilan Data *Real-Time* dari *Firestore Realtime Database*, Mengukur kecepatan dan respon aplikasi dalam menampilkan data sensor secara *real-time* dan historis.

B. Hasil Pengujian

1. Pengujian Fungsional Aplikasi

Setelah melakukan uji coba di aplikasi dengan beberapa percobaan yaitu percobaan *sign up* sebanyak 3 kali dengan menggunakan *username* dan *password* yang berbeda dan percobaan *login* sebanyak 10 kali untuk satu *username* dan *password* lalu mengakses halaman *realtime*, *history average weather A*, *history average weather B*, *history per minutes weather A*, *history per minutes weather B* serta *side menu search history-history* sebanyak 30 kali setiap satu *username* dan *password* 10 kali percobaan untuk mengetahui apakah aplikasi ini berjalan dengan baik atau mengalami gangguan atau error.

TABEL 1. Uji Fungsional Aplikasi

No	Skenario Pengujian	Aksi	Hasil Yang Diharapkan	Persentase Keberhasilan
1	Masuk Aplikasi	Klik Aplikasi Skycast	Masuk ke menu halaman awal aplikasi	100%
2	Sign Up dan Sign In	Klik Button Sign Up atau Sign In	Dapat mengisi <i>username</i> dan <i>password</i> untuk proses <i>sign up</i> dan <i>sign in</i> untuk <i>login</i> ke	100%

			aplikasi	
3	Melihat Data <i>Realtime</i>	Klik <i>Button Realtime</i> dan akan masuk ke <i>screen realtime</i>	User dapat melihat hasil data <i>realtime</i> dari sensor suhu, kelembaban, raindrop dan tekanan	100%
4	Melihat <i>history average weather A</i> dan <i>weather B</i>	Klik <i>button history average weather A</i> atau <i>weather B</i> dan akan masuk ke <i>screen history average weather A</i> atau <i>weather B</i>	User dapat melihat <i>history average weather A</i> atau <i>weather B</i> yang menampilkan data tiap 30 menit	90% (Gangguan pada saat menampilkan data <i>history</i> dikarenakan data pada <i>firebase</i> terlalu banyak)

8	Saya merasa mudah mengakses aplikasi ini untuk pertama kali.
9	Saya merasa mudah memahami data yang ditampilkan dalam aplikasi ini.
10	Saya merasa nyaman menggunakan aplikasi ini secara keseluruhan.

Pada penelitian ini menggunakan metode kuantitatif yang didapatkan hasilnya dari para pengguna yang mengisi *kuesioner Usability*. Setelah data terkumpul hasilnya kemudian digunakan untuk dianalisis tingkat efektivitas aplikasi serta tingkat kepuasan pengguna dalam menggunakan aplikasi tersebut. Setelah data terkumpul dari para responden, kemudian diakumulasikan. Pada penelitian [5] untuk perhitungan *usability* ini memiliki aturan :

1. Tiap pertanyaan ganjil, skor dari nilai asli dikurangi 1
2. Tiap pertanyaan genap, skor dari nilai 5 dikurangkan dengan skor dari responden
3. Semua hasil skor didapat dari akumulasi tiap pertanyaan yang kemudian dikali 2,5

Aturan ini hanya berlaku untuk 1 pengguna. Untuk akumulasi selanjutnya, skor *usability* dari setiap responden dicari rata-rata skornya dengan menambahkan semua skor lalu dibagi dengan jumlah responden. Berikut persamaannya (1.1):

$$\bar{x} = \frac{\sum x}{n}$$

dimana \bar{x} merupakan nilai rata-rata dari *usability*, $\sum x$ merupakan jumlah nilai *usability*, dan n merupakan jumlah responden.

Untuk mengetahui tingkat efektivitas dapat dilihat pada tabel 3.

TABEL 3.
Tingkat Efektivitas

No	Perbandingan Efektivitas	Capaian
1	< 40%	Sangat Tidak Efektif
2	40% - 59,99%	Tidak Efektif
3	60% - 79,99%	Efektif
4	≥ 80%	Sangat Efektif

2. Pengujian Aplikasi Dengan *System Usability Scale (SUS)*.

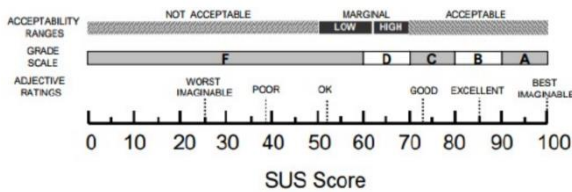
Menurut penelitian [4], *System Usability Scale (SUS)* adalah alat evaluasi yang dipakai untuk menilai sejauh mana kegunaan sebuah sistem. Diperkenalkan oleh John Brooke pada tahun 1986, *SUS* dapat diterapkan untuk mengukur kegunaan berbagai macam produk, termasuk perangkat keras, perangkat lunak, aplikasi seluler, dan situs web. *System Usability Scale* terdiri dari 10 pernyataan yang dinilai menggunakan *skala Likert* dengan lima tingkat pilihan jawaban, mulai dari sangat tidak setuju hingga sangat setuju. Tabel 2 berisi 10 pertanyaan yang ditanyakan pada *kuesioner*.

TABEL 2.
Pertanyaan Kuesioner

No	Pertanyaan
1	Desain visual aplikasi ini baik.
2	Penggunaan warna dalam aplikasi ini membantu memperjelas informasi.
3	Ikon dan gambar dalam aplikasi ini membantu saya dalam navigasi.
4	Tombol-tombol dalam aplikasi ini mudah ditemukan dan diakses.
5	Penyajian data dalam aplikasi ini jelas.
6	Saya merasa fitur-fitur yang saya butuhkan mudah ditemukan dalam aplikasi ini.
7	Saya merasa mudah menavigasi aplikasi ini.

Menurut penelitian [6], Analisis kepuasan pengguna dilakukan melalui *usability testing*, yang diukur menggunakan *kuesioner System Usability Scale (SUS)* yang dibagikan kepada pengguna. *Kuesioner usability* terdiri dari 10 pertanyaan, dengan setiap pertanyaan menggunakan *Skala Likert* untuk mengukur sikap, pendapat, dan persepsi individu atau kelompok. Dalam studi ini, *Skala Likert* memiliki 5 level penilaian, yaitu: sangat tidak setuju, tidak setuju, netral, setuju, dan sangat setuju. Hasil dari pengukuran *kuesioner* tersebut kemudian dianalisis dengan membandingkan skor

SUS terhadap standar yang telah ditentukan.



GAMBAR 10.
Skor SUS

Setelah data terkumpul, langkah berikutnya adalah menghitung setiap pernyataan ganjil dan genap pada kuesioner. Peneliti kemudian menjumlahkan nilai dari setiap pertanyaan. Skor maksimum untuk tiap pertanyaan adalah 5, sedangkan skor minimum adalah 0. Setelah menjumlahkan skor dari 10 pernyataan tersebut, langkah selanjutnya adalah mengalikan totalnya dengan 2,5. Hasil dari perhitungan ini akan menjadi SUS Skor untuk aplikasi Android Skycast.

Sesuai dengan hasil tabel yang didapatkan nilainya, dapat diketahui bahwa aplikasi Skycast untuk memantau cuaca berhasil menarik perhatian para pengguna aplikasi ini. Hasil dari rata-rata skor SUS adalah 71%, aplikasi ini mampu bekerja dengan “efektif”. Ini menunjukkan Skycast dianggap efektif dalam memenuhi kebutuhan pengguna untuk memantau cuaca secara realtime. Dengan menggunakan aplikasi Skycast, pengguna merasa lebih nyaman dan efisien dalam mendapatkan informasi cuaca yang mereka butuhkan. Meskipun demikian, terdapat beberapa aspek yang bisa diperbaiki untuk mencapai skor yang lebih tinggi dan memberikan pengalaman pengguna yang lebih optimal.

C. Analisis Hasil Pengujian

Pengujian aplikasi android dapat dilihat pada tabel pengujian fungsional aplikasi dengan melihat fungsionalitas pada aplikasi. Terdapat beberapa skenario pengujian untuk memastikan bahwa fitur pada aplikasi telah dikembangkan dengan baik, memastikan *firebase authentication* mengsetting data pengirim, *firebase realtime database* menyimpan data yang diterima dari sensor suhu, kelembaban, raindrop dan tekanan secara *real-time*, memverifikasi Koneksi dan Integrasi Aplikasi dengan *Firestore Realtime Database*, bahwa aplikasi terintegrasi dengan *Firestore Realtime Database* untuk penyimpanan dan pengambilan data sensor, pengujian Kecepatan dan respon Aplikasi terhadap Perubahan Data Sensor serta Pengambilan Data *Real-Time* dari *Firestore Realtime Database*, mengukur kecepatan dan respon aplikasi dalam menampilkan data sensor secara *real-time* dan historis. Dengan keberhasilan pengujian aplikasi android ini berdasarkan status pengujian yang sesuai dari seluruh skenario yang dilakukan.

Dengan keberhasilan pengujian aplikasi android Skycast berdasarkan data status pengujian yang sesuai dari seluruh skenario yang dilakukan, dapat disimpulkan bahwa aplikasi ini tidak hanya dapat memenuhi kebutuhan pengguna dalam hal kegunaan tetapi juga berfungsi dengan baik secara teknis. Pengguna merasa nyaman menggunakan aplikasi ini, dan semua fitur sesuai dengan harapan para pengguna. Integrasi dengan *Firestore Realtime Database* berjalan lancar, dan aplikasi mampu menampilkan data sensor dengan cepat dan akurat. Meskipun demikian, ada ruang untuk peningkatan

lebih lanjut guna mencapai skor SUS yang lebih tinggi dan memberikan pengalaman pengguna yang lebih unggul.

V. KESIMPULAN

Dapat disimpulkan bahwa aplikasi Skycast telah mencapai tujuan utamanya sebagai aplikasi monitoring cuaca secara *realtime* yang efisien, serta dapat menyediakan akses riwayat informasi cuaca dengan rentang waktu per menit dan per 30 menit. Dengan diintegrasikan dengan sistem IoT dan *cloud* yang cukup kuat, aplikasi ini memberikan akses data yang akurat dan *up-to-date* bagi para pengguna. Dengan melakukan pengujian fungsional aplikasi dimana pada pengujian ini bisa dikatakan 90% berhasil dengan kata lain aplikasi ini dapat berfungsi dengan baik dan untuk pengujian dengan metode *System Usability Scale (SUS)* dengan skor yang didapatkan sebesar 71% menunjukkan bahwa aplikasi ini efektif digunakan untuk memonitoring cuaca secara *realtime*. meskipun memenuhi kebutuhan para pengguna dengan baik, masih ada beberapa aspek yang perlu ditingkatkan untuk memberikan pengalaman baik kepada pengguna di masa yang akan datang. Namun secara keseluruhan, Aplikasi Skycast adalah solusi yang efektif dan praktis untuk memantau kondisi cuaca secara *realtime* dengan potensi pengembangan yang lebih lanjut

REFERENSI

- [1] S. A. Santoso, "Monev (3): Pengumpulan Data Monitoring," **Integrasi Edukasi**, Dec. 3, 2015. [Online]. Available: <https://www.integrasi-edukasi.org/monev-3-pengumpulan-data-monitoring/>. [Accessed: Aug. 21, 2024].
- [2] A. Juansyah, "Pembangunan aplikasi child tracker berbasis assisted-global positioning system (A-GPS) dengan platform Android," **Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)**, vol. 1, no. 1, pp. 1-9, Aug. 2015, ISSN: 2089-9033.
- [3] Universitas Aisyiah Yogyakarta, "Apa Itu MIT App Inventor, Berikut Penjelasannya," **Program Studi Teknologi Informasi Universitas Aisyiah Yogyakarta**, Jan. 6, 2020. [Online]. Available: <https://psti.unisayogya.ac.id/2020/01/06/apa-itu-mit-app-inventor-berikut-penjelasannya/>. [Accessed: Aug. 20, 2023].
- [4] J. Brooke, "SUS: A quick and dirty usability scale," **Usability Evaluation in Industry**, 1996. [Online]. Available: <https://www.researchgate.net/publication/228593520>
- [5] Sharfina, Z., & Santoso, H. B. (2017). An Indonesian adaptation of the System Usability Scale (SUS). 2016 International Conference on Advanced Computer Science and Information Systems,

ICAC SIS 2016, 145–148.
<https://doi.org/10.1109/ICAC SIS.2016.7872776>

[6] E. Susilo, "Cara menggunakan System Usability Scale (SUS) pada evaluasi usability," UI/UX |

Usability, 7 March 2019. [Online]. Available:
<https://www.edisusilo.com/cara-menggunakan-system-usability-scale/> . Accessed: Aug. 4, 2024.

