

Sistem Pemantauan Parkir Liar Menggunakan Sensor Infrared Berbasis Android (Studi Kasus Politeknik Negeri Malang)

Nur Ahmad Wasian¹, Mila Kusumawardani², Ridho Hendra Yoga Perdana³

¹ Program Studi Jaringan Telekomunikasi Digital,
Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia
^{2,3} Program Studi Teknik Telekomunikasi,
Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

supiahmad420@gmail.com, mila.kusumawardani@polinema.ac.id, ridho.hendra@polinema.ac.id

Abstract—Illegal parking in Malang State Polytechnic is still not well monitored. The Officer Pengamanan Dalam(Pamdal), whose job is to curb Polinema residents who park their vehicles, cannot monitor the vehicles regularly, so there are still many who park illegally. Based on this, an automatic system that can be used by Pamdal to monitor illegal parking in Polinema is needed. Therefore, "Illegal Parking Monitoring Information System Using Infrared Sensor Based On Android" was created. The system built uses several components, including Arduino Nano, ESP32 Cam, Infrared Sensor, and servo motor. With the making of this tool and research, it is hoped that it will be able to assist in disciplining polynemaic residents to order parking through monitoring of the system being built. Three pieces of equipment were made and placed in front of the AH Building, in front of the AI Building, and next to the AO Building. The results of the experiment for notifications from the three nodes have different delays in sending data from the node to the smartphone on the user side. Delay at node two has the greatest delay, which is 193.0708 ms. Meanwhile, node three has excellent performance with an average delay value of 107.0723 ms.

Keywords: Parking, PAMDAL, Infrared Sensor, ESP32-CAM, Android.

Abstrak—Parkir liar yang berada di Politeknik Negeri Malang masih belum terpantau dengan baik. Petugas Pengamanan Dalam(Pamdal) yang bertugas untuk menertibkan warga Polinema memarkirkan kendaraan tidak dapat memantau kendaraan secara berkala, sehingga masih banyak yang melakukan parkir liar. Berdasarkan hal tersebut diperlukan sistem otomatis yang dapat digunakan Pamdal dalam memantau parkir liar di Polinema. Oleh karena itu dibuat "Sistem Informasi Pemantauan Parkir Liar Menggunakan Sensor Infrared Berbasis Android". Sistem yang dibangun menggunakan beberapa komponen, diantaranya Arduino Nano, ESP32 Cam, Sensor Infrared, dan motor servo. Dengan dibuatnya perangkat dan penelitian ini diharapkan mampu membantu pendisiplinan warga polinema untuk tertib parkir melalui pemantauan dari sistem yang dibangun. Perangkat dibuat sebanyak tiga buah dan diletakkan di depan Gedung AH, di depan Gedung AI, dan di samping Gedung AO. hasil dari percobaan untuk notifikasi dari ketiga *node* memiliki delay yang berbeda pengiriman datanya dari *node* ke smartphone pada sisi user. Untuk *Delay* di *node* dua memiliki *delay* paling besar yakni sebesar 193.0708 ms. Sedangkan untuk *node* tiga memiliki performansi sangat baik dengan nilai rata-rata *delay* sebesar 107.0723 ms.

Kata kunci: Parkir, PAMDAL, Sensor Infrared, ESP32-CAM, Android.

I. PENDAHULUAN

Politeknik Negeri Malang (Polinema) adalah perguruan tinggi negeri yang terdapat di Kota Malang. Polinema memiliki banyak lahan parkir yang berada di beberapa lokasi dengan kapasitas 4.250 untuk kendaraan roda dua dan 250 untuk kendaraan roda empat. Lokasi parkir untuk di Polinema terletak di dekat masing-masing Gedung. Namun lokasi area parkir tersebut seringkali tidak digunakan oleh pengendara sebagaimana fungsinya. Banyak kendaraan roda dua parkir

tidak pada lahan yang telah disediakan dan memilih melakukan parkir ditempat yang tidak seharusnya. Parkir liar yang dilakukan pengendara mengakibatkan terganggunya kepentingan umum dan mengganggu pejalan kaki. Tempat yang sering digunakan untuk parkir liar antara lain di depan Gedung AH, di depan Gedung AI, dan di sebelah Gedung AO. Selama ini, parkir liar yang berada di Polinema masih belum terpantau dengan baik. Pengamanan Dalam(Pamdal) yang bertugas untuk menertibkan warag Polinema memarkirkan kendaraan belum

mampu melakukan tugas dengan maksimal, sehingga masih banyak warga Polinema yang melanggar area parkir. Selain itu tidak terdapat sistem kontrol terhadap pemantauan kendaraan mahasiswa, dosen, dan staf yang dapat dijadikan bahan evaluasi perparkiran. Sehingga seharusnya area yang seharusnya kosong karena dilarang digunakan untuk parkir justru penuh dengan kendaraan yang memarkirkan kendaraannya sembarangan.

Pada penelitian sebelumnya menjelaskan tentang aplikasi yang dapat menyajikan log parkir dalam kurun waktu yang diinginkan menggunakan Sensor Ultrasonik sebagai penangkap objek. Namun penggunaan sensor ultrasonik untuk mengukur jarak keberadaan objek kurang presisi[1].

Penelitian selanjutnya merancang dan merealisasikan model sistem monitoring perparkiran dengan fasilitas pemilihan area parkir dengan berbasis Raspberry Pi serta pemanfaatan infrared sebagai sensor. Namun pada penelitian ini memiliki kelemahan penyalarsan Raspberry Pi dengan perangkat yang digunakan[2]. Kemudian selain itu penelitian berjudul “Rancang Bangun Monitoring Parkir Berbasis Arduino” membahas tentang sistem yang dibuat untuk mempermudah pelanggan untuk mencari tempat parkir yang masih kosong sehingga mengurangi terjadi menyebabkan antrian dan kemacetan sehingga mengganggu kelancaran lalu lintas. Namun penggunaan sensor ultrasonic dinilai kurang akurat untuk mendeteksi jarak objek[3].

Penelitian selanjutnya dijelaskan mengenai suatu sistem dan diimplementasikan pada salah satu lokasi parkir, yaitu khusus untuk parkir dosen dan karyawan. Dengan memanfaatkan teknologi RFID, Arduino, motor stepper, sensor IR. Sistem masih dalam proses, terutama untuk bagian aktuator/ penggerak palang portal[4].

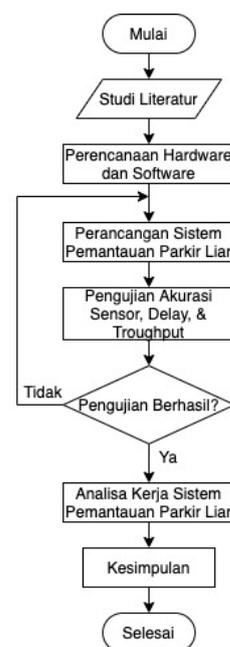
Penelitian selanjutnya menyebutkan bahwa perancangan system menggunakan mikrokontroler sebagai pengendali utama. Untuk tiap-tiap lokasi parkir mobil, dipasang sensor infra merah yang akan memberikan informasi dilokasi itu terdapat kendaraan/mobil atau tidak. Dari hasil pengolahan keluaran sensor tadi akan diberikan informasi berupa display yang akan menunjukkan keadaan jumlah kapasitas tempat parkir yang sudah terisi dan sisanya. Namun penggunaan mikrokontroler AT89C51 perlu diganti dengan perangkat yang lebih modern[5].

Oleh karena itu dibuat “**Sistem Pemantauan Parkir Liar Menggunakan Sensor Infrared Berbasis Android**”. Sistem yang akan dibangun menggunakan beberapa komponen, diantaranya Arduino Nano, ESP32 Cam, Sensor Infrared, dan motor servo. Dengan dibuatnya perangkat dan penelitian ini diharapkan mampu membantu pendisiplinan warga polinema untuk tertib parkir melalui pemantauan dari sistem yang dibangun. Perangkat dibuat sebanyak tiga(3) buah dan diletakkan di depan Gedung AH, di depan Gedung AI, dan di samping Gedung AO.

II. METODE

A. Rancangan Penelitian

Pada penelitian ini akan dibahas tentang perancangan sistem pemantauan parkir liar. Proses perancangan ini melalui beberapa tahapan sebagai berikut:



Gambar 1. Rancangan penelitian

Gambar 1 merupakan rancangan penelitian yang akan dilakukan, Tahapan pertama yang dilakukan yaitu:

1) *Studi literatur*: Sebelum melakukan penelitian terlebih dahulu mencari referensi mengenai informasi penunjang dalam penelitian yang dilakukan. Referensi penunjang meliputi alat dan bahan yang akan digunakan.

2) *Perencanaan hardware dan software sistem*: dalam perencanaan ini meliputi desain perangkat, sistem output data dan perkiraan letak perangkat yang diperlukan.

3) *Perancangan sistem*: berupa perencanaan alat dan perencanaan mengenai alur saat penelitian. Perencanaan yang akan dibuat pada penelitian ini ditunjukkan pada blok diagram system secara keseluruhan.

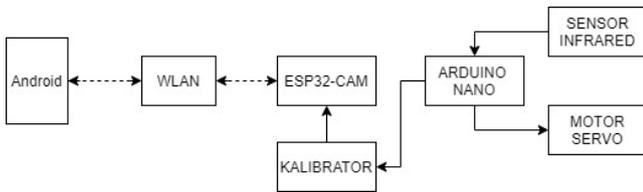
4) *Pengujian sistem*: sistem yang telah dibuat akan dilakukan pengujian dan analisa, meliputi pengujian jarak sensor, pengujian modul kamera, pengujian program, dan pengujian pengiriman data ke server untuk selanjutnya dilakukan analisa sistem.

5) *Analisa sistem*: pada tahap ini dilakukan analisa sistem kerja diantaranya program dapat berjalan sesuai dengan sistem yang telah direncanakan, pengiriman data ke server, sensitivitas sensor yang digunakan.

6) *Hasil sistem kerja*: jika hasil sistem kerja yang diperoleh memenuhi sistem yang telah direncanakan maka dapat ditarik kesimpulan, apabila hasil sistem kerja yang diperoleh tidak memenuhi sistem yang telah direncanakan maka akan dilakukan pengecekan ulang pada perancangan *hardware* dan *software*, perancangan desain dan pengecekan pada pembuatan program di arduino Uno yang telah dibuat.

7) *Kesimpulan*: pembuatan kesimpulan ini dapat dilakukan apabila hasil dari sistem kerja telah sesuai dengan perancangan sistem yang telah dibuat.

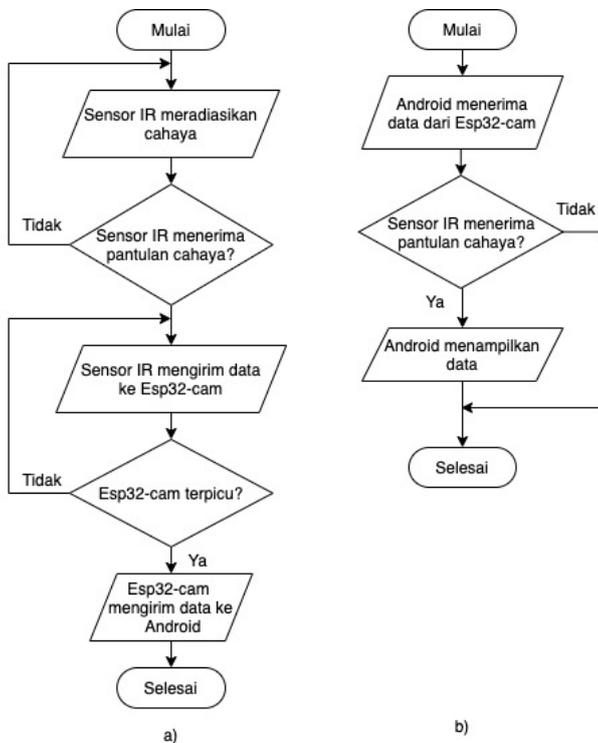
B. Blok Diagram Sistem



Gambar 2. Blok diagram

Pada gambar di atas menjelaskan tentang alur yang dilakukan pada alat monitoring parkir liar dimana, sensor *Infrared* akan mendeteksi kendaraan yang melakukan parkir liar yang kemudian akan mengirimkan data kepada Arduino nano yang kemudian dikirim ke Esp32-cam melalui Kalibrator sebagai pemicu Esp32-cam, dan setelah menerima data dari Arduino nano, kemudian Esp32-cam akan mengirimkan data berupa gambar ke server *database*. Kemudian Android yang telah terinstall aplikasi monitoring parkir liar menerima data dari *database* server dan menampilkan di *interface* Android.

C. Flowchart



Gambar 3. Flowchart

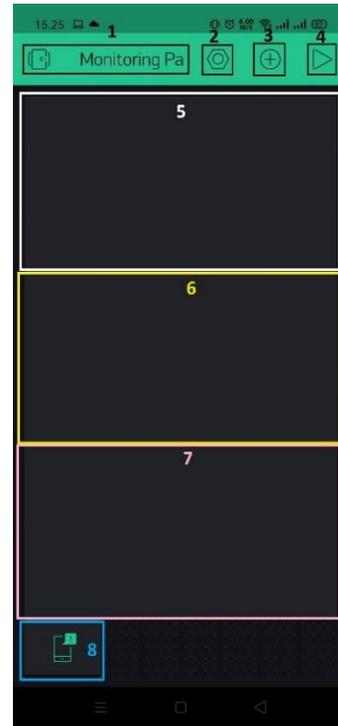
Sensor *infrared* mendeteksi kendaraan yang melakukan parkir liar, jika kendaraan yang melakukan parkir liar berhasil terdeteksi maka sensor *infrared* akan mengirim data pada Esp32-cam, jika kendaraan yang melakukan parkir liar gagal dideteksi oleh sensor *infrared*, maka sensor *infrared* akan mendeteksi ulang. Setelah Esp32-cam menerima data dari sensor *infrared*, Esp32-cam akan dipicu oleh kalibrator untuk mengambil gambar kendaraan yang melakukan parkir liar, kemudian mengirim data berupa gambar ke aplikasi di android dan menyimpan gambar di SD Card pada Esp32-cam.

Aplikasi pada android menerima data dari Esp32-cam, android akan menampilkan data dari Esp32-cam dalam bentuk

notifikasi dan gambar pada aplikasi. jika tidak, maka tidak ada data yang akan ditampilkan.

D. Perancangan Aplikasi Blynk

Pada Gambar di bawah menampilkan tampilan aplikasi monitoring melalui android, aplikasi tersebut dibuat dengan menggunakan Aplikasi Blynk yang telah diintegrasikan dengan mikrokontroller yang digunakan sebagai penampilan datanya, lalu pada data yang tersimpan tersebut ditampilkan pada aplikasi android yang telah dibuat seperti Gambar, tampilan yang dihasilkan berupa data gambar dan teks. Berikut proses pembuatan aplikasi yang digunakan:



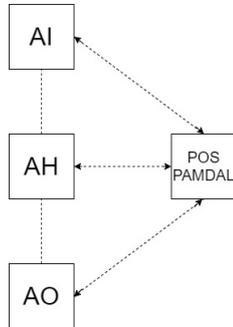
Gambar 4. Perancangan aplikasi blynk

Pada gambar di atas menunjukkan perancangan aplikasi *Blynk* yang akan digunakan untuk memantau parkir liar. Untuk menu yang disimbolkan oleh nomor satu (1) menunjukkan simbol yang digunakan untuk keluar dari menu aplikasi *monitoring*. Untuk nomor dua (2) menunjukkan menu untuk mengatur aplikasi. Kemudian simbol nomor tiga (3) untuk menambah fitur yang diinginkan. Pada nomor empat (4) menunjukkan simbol *play* untuk memulai system pada aplikasi. Kotak nomor lima (5) merupakan area pada aplikasi untuk menunjukkan pemantauan pada area pertama yaitu depan Gedung AH. Kotak nomor lima (6) merupakan area pada aplikasi untuk menunjukkan pemantauan pada area kedua yaitu depan Gedung AI. Kotak nomor lima (7) merupakan area pada aplikasi untuk menunjukkan pemantauan pada area ketiga yaitu sebelah Gedung AO. Terakhir merupakan *button* untuk mengatur notifikasi pada system aplikasi *Blynk* ditunjukkan pada angka delapan (8).

E. Denah Area Pengujian Alat

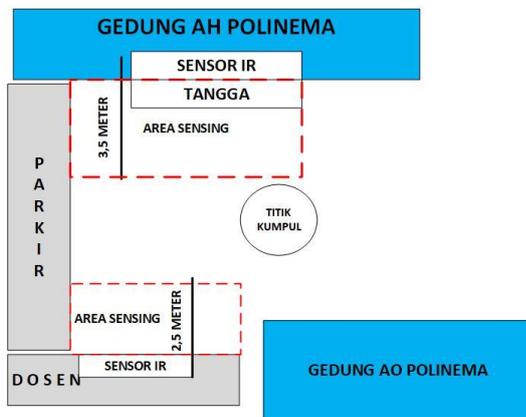
Sebelum melakukan pengujian perangkat yang sudah dirancang, dibuat desain desain denah area yang akan dijadikan

sebagai *sensing area* dalam penelitian ini. Desain yang digambar adalah area parkir liar di depan Gedung AH depan Gedung AH dan depan gedung AI. Pada gambar 5 merupakan skema alur pengiriman data dari tiga lokasi ke aplikasi android yang berada di pos pamdal. Pada saat alat pemantauan parkir liar pada Gedung AI, AH, dan AO mendeteksi adanya parkir liar maka alat pemantauan akan mengirimkan data dan diterima oleh aplikasi android yang terletak di pos pamdal.



Gambar 5. Skema alur pengiriman data

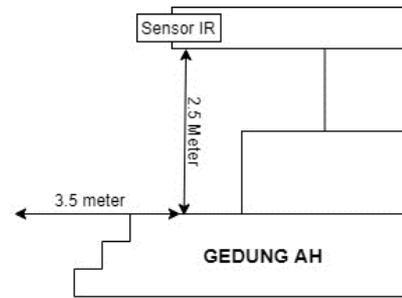
Sensing area yang diuji dari letak sensor IR memiliki jarak yang berbeda-beda. Untuk *sensing area* yang berada di depan Gedung AH sejauh 3.5 meter, kemudian untuk *sensing area* di samping Gedung AO sejauh 2 meter, dan untuk *sensing area* yang berada di depan Gedung AI sejauh 4 meter. Jarak dari masing-masing *sensing area* tersebut telah diukur menggunakan alat ukur berupa meteran dan tidak lebih dari jarak yang ditentukan karena sensor hanya mampu mendeteksi hingga 500 cm. Denah yang digunakan untuk pengujian di depan Gedung AH dan samping Gedung AO ditunjukkan oleh gambar 6, 8 dan 9. Sedangkan denah yang digunakan untuk pengujian di depan Gedung AI ditunjukkan oleh gambar 7 dan 10.



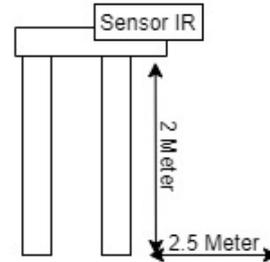
Gambar 6. Denah Sensing Area Gedung AH dan AO



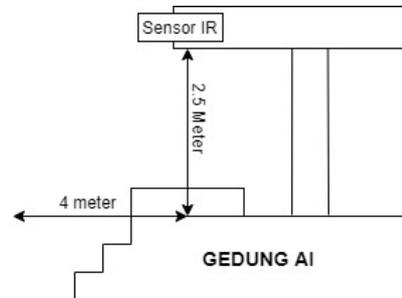
Gambar 7. Denah sensing area gedung AI



Gambar 8. Sensing area gedung AH tampak samping

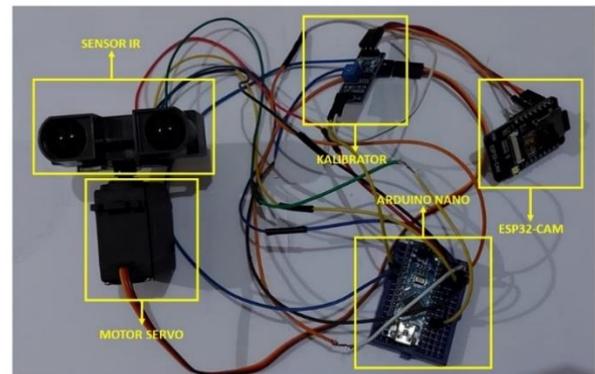


Gambar 9. Sensing area gedung AO tampak samping



Gambar 10. Sensing area gedung AI tampak samping

F. Implementasi Alat



Gambar 11. Implementasi alat

Pada gambar di atas menampilkan implementasi alat monitoring melalui android, alat tersebut dibuat sebagai mikrokontroler yang digunakan sebagai penampikan datanya, lalu pada data yang tersimpan tersebut ditampilkan pada aplikasi android. Implementasi alat pada penelitian ini terdapat 3 perangkat untuk dipasang pada tiga tempat yang berbeda, tetapi memiliki rangkaian yang sama. Terdiri atas Arduino

nano, Motor Servo, sensor *Infrared*, Esp32-cam, dan Kalibrator.

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Akurasi Sensor

TABEL I
HASIL PENGUJIAN SENSOR NODE 1

Sensor Infrared		
Jarak(cm)	Deteksi	Δ
120	120	0 cm
280	280	0 cm
380	381	1 cm
400	399	1 cm
420	-	-

Dapat dilihat pada tabel I bahwa dari 16 kali percobaan pada node 1 yang berada di depan gedung AH, dari jarak 120 sampai dengan 400 cm hanya mendapatkan selisih 1 cm di jarak 380 cm dan 400 cm. Pada jarak 120 cm hingga 360 cm sensor menunjukkan bahwa memiliki akurasi 100% dengan selisih pengukuran 0 cm.

TABEL II
HASIL PENGUJIAN SENSOR NODE 2

Sensor Infrared		
Jarak(cm)	Deteksi	Δ
120	120	0 cm
280	280	0 cm
380	380	0 cm
400	399	2 cm
420	-	-

Pada pengujian *node 2* dengan 16 kali percobaan terdapat satu kali kegagalan deteksi pada jarak 400 cm dengan selisih 2 cm. Berbeda dengan node 1 yang memiliki selisih pada jarak 380 cm dan 400 cm.

TABEL III
HASIL PENGUJIAN SENSOR NODE 3

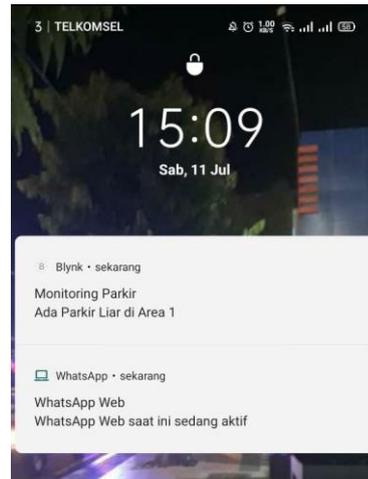
Sensor Infrared		
Jarak(cm)	Deteksi	Δ
120	120	0 cm
280	280	0 cm
380	380	0 cm
400	399	1 cm
420	-	-

Pada pengujian deteksi objek untuk *node 3* yang dipasang di sebelah Gedung AO. Dari data yang ditunjukkan oleh tabel 4.3 dapat dilihat bahwa pada jarak 100 cm sampai dengan 380 cm sensor mampu mendeteksi objek dengan baik dan hanya memiliki selisih 1 cm pada jarak 400. Dari ketiga tabel di atas dapat dilihat bahwa pada jarak 420 cm Sensor *Infrared* tidak dapat mendeteksi objek dengan baik.

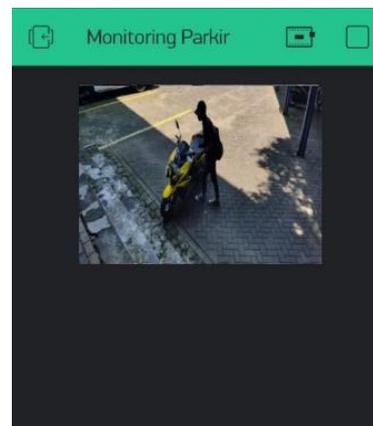
B. Hasil Tampilan Aplikasi

Setelah pengujian kemampuan deteksi dari node yang dibuat, berikutnya adalah tampilan aplikasi ketika tidak ada objek dan sesudah ada objek. Aplikasi ini dibuat untuk memudahkan melihat gambar yang tertangkap oleh node.

Diharapkan dengan adanya aplikasi ini dapat digunakan pengguna dalam memantau parkir liar yang sesuai dengan tujuan penelitian yang dibuat. Berikut adalah gambar tampilan pada android dan aplikasi yang digunakan:



Gambar 12. Tampilan notifikasi aplikasi



Gambar 13. Tampilan pada aplikasi

C. Troughput

Pengukuran throughput dilakukan pada bagian ini diperlukan untuk melihat berapa kecepatan yang digunakan untuk mengirimkan data gambar dari node ke aplikasi. Hasil pengukuran dapat dilihat pada tabel di bawah ini:

TABEL I.
PENGUKURAN TROUGHPUT NODE 1 AH

Hari Ke	Perhitungan Throughput Node 1 AH	
	Total paket (Mbytes)	Throughput (kbps)
1	12531.25	3133.721
2	12631.33	3233.642
3	12424.63	3013.102
Rata- Rata	12736.78833	3223.102167

TABEL II.
PENGUKURAN THROUGHPUT NODE 2 AO

Hari Ke	Perhitungan Throughput Node 2 AO	
	Total paket (Mbytes)	Throughput (kbps)
1	12621.31	3233.621
2	12844.71	3347.712
3	12952.12	3651.114
Rata- Rata	12746.37	3322.133333

TABEL III.
PENGUKURAN THROUGHPUT NODE 3 AI

Hari Ke	Perhitungan Throughput Node 3 AI	
	Total paket (Mbytes)	Throughput (kbps)
1	12231.91	3243.132
2	12314.82	3821.116
3	12634.32	3562.325
Rata- Rata	12513.00667	3486.244333

Berdasarkan tabel di atas, bisa dilihat bahwa node 1 memiliki pengiriman gambar 3223.102167 kbps, sedangkan untuk node 2 memiliki kecepatan pengiriman gambar sebesar 3322.133333 kbps dan untuk node 3 memiliki kecepatan pengiriman gambar sebesar 3486.244333 kbps. Disini node 3 paling besar karena kontras warna yang ditangkap cukup tinggi. Sehingga data yang dikirimkan lebih besar dan throughput menjadi tinggi.

D. Delay Pengiriman Informasi

TABEL IV.
PENGUKURAN DEAY NODE 1

Ip Source	Ip Destination	Delay
192.168.100.62	192.168.100.15	0.001467 s
192.168.100.62	192.168.100.15	0.240107 s
192.168.100.62	192.168.100.15	0.330282 s
192.168.100.62	192.168.100.15	0.041931 s
Rata-Rata		0.1367185 s

TABEL V.
PENGUKURAN DELAY NODE 2

Ip Source	Ip Destination	Delay
192.168.100.62	192.168.100.15	0.245398 s
192.168.100.62	192.168.100.15	0.000125 s
192.168.100.62	192.168.100.15	0.104121 s
192.168.100.62	192.168.100.15	0.606030 s
Rata-Rata		0.1930708 s

TABEL VI.
PENGUKURAN DELAY NODE 3

Ip Source	Ip Destination	Delay
192.168.100.62	192.168.100.15	0.195298 s
192.168.100.62	192.168.100.15	0.182876 s

Ip Source	Ip Destination	Delay
192.168.100.62	192.168.100.15	0.003121 s
192.168.100.62	192.168.100.15	0.002994 s
Rata-Rata		0.1070723 s

Dari tabel diatas dapat dilihat bahwa rata rata delay yang dimiliki oleh node 1 dan node 3 termasuk dalam kategori yang sangat bagus. Hal ini dikarenakan node 1 memiliki rata rata delay sebesar 0.1367185 s atau 136.718 ms (dibawah 150 ms). Sementara untuk node 3 memiliki rata rata delay sebesar 0.1070723 atau 107.072 ms (dibawah 150 ms). Sementara untuk node 2 memiliki kategori yang baik dengan rata-rata delay 0.1930708 atau 193.070 ms(di atas 150 ms).

IV. KESIMPULAN

Sensor Infrared yang digunakan dalam pengujian ini berhasil mendeteksi adanya kendaraan dengan memancarkan sinar inframerah, jarak pengukuran maksimal 4 meter dari titik pemasangan sensor hingga *sensing area*. Esp32-cam mengambil gambar dengan delay camera 3-4 detik dan mengirim ke aplikasi pada android, untuk pengiriman gambar ke aplikasi pada ketiga node memiliki throughput yang berbeda dimana node 2 memiliki throughput yang besar yaitu 3263817 kbps yang disebabkan gambar memiliki kontras warna yang tinggi (efek terkena sinar matahari). Aplikasi berhasil menampilkan notifikasi dan gambar objek yang melakukan parkir liar, untuk notifikasi dari ketiga *node* memiliki delay yang berbeda pengiriman datanya dari *node* ke smartphone pada sisi user. Untuk *Delay* di *node* dua memiliki *delay* paling besar yakni sebesar 193.070 ms. Sedangkan untuk *node* tiga memiliki performansi paling baik dengan nilai rata-rata *delay* sebesar 107.072 ms.

REFERENSI

- [1] A. B. Warsito, M. Yusup, and M. Aspuri, "Penerapan Sistem Monitoring Parkir Kendaraan Berbasis Android Pada Perguruan Tinggi Raharja," *Technomedia J.*, vol. 2, no. 1, pp. 82–94, 2017, doi: 10.33050/tmj.v2i1.317.
- [2] D. NATALIANA, I. SYAMSU, and G. GIANTARA, "Sistem Monitoring Parkir Mobil menggunakan Sensor Infrared berbasis RASPBERRY PI," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 2, no. 1, p. 68, 2014, doi: 10.26760/elkomika.v2i1.68.
- [3] M. Zikri and R. Khair, "Rancang Bangun Monitoring Parkir Berbasis Arduino," *Teknovasi*, vol. 05, no. 1, pp. 27–38, 2018.
- [4] N. H. Sudibyo, L. Rosmalia, and A. Sofyan, "Analisis Dan Implementasi Sistem Parkir Kendaraan Bermotor (Studi Kasus Institut Informatika Dan Bisnis Darmajaya)," pp. 114–119, 2019.
- [5] Rahmani, A. S. B. Nugroho, and budi Rahmani, "Sistem Informasi Parkir Menggunakan Sensor Infra Merah Terkendali Mikrokontroler At89C51," *J. Progresif*, vol. 4, no. 1, pp. 401–416, 2008, doi: 10.1093/jac/dkl490.