



## **TECHNES: Penyedia Layanan Jasa Service Berbasis Aplikasi Android**

<sup>1\*</sup>M Syahid Nur Wahid, <sup>2</sup>Ali Mirza Najafi, <sup>3</sup>Alfaiz Sihaj, <sup>4</sup>Rahel Alik, <sup>5</sup>Asisah Az Zahrah Irsani

<sup>12345</sup>Universitas Negeri Makassar, Jl. Mallengkeri Raya, Parangtambung, Kec. Tamalate, Kota Makassar, Sulawesi Selatan 90224, Indonesia

Email: <sup>1\*</sup>syahid@unm.ac.id, <sup>2</sup>najafialimirza@gmail.com, <sup>3</sup>faizsija21@gmail.com, <sup>4</sup>rahelalik246@gmail.com, <sup>5</sup>azizahzahrahirsani.03@gmail.com

### **ABSTRAK**

Artikel ini menjelaskan bagaimana pendekatan metode *waterfall* digunakan untuk pembuatan aplikasi Android oleh *Technes*, penyedia layanan berbasis aplikasi. Metode *waterfall* adalah pendekatan *metodis* dan *linier* untuk mengembangkan perangkat lunak yang dimulai dengan menganalisis persyaratan dan berkembang melalui desain, pengujian, dan implementasi. Pada artikel ini, kita akan membahas bagaimana metode *waterfall* diterapkan pada pengembangan aplikasi *Technes* untuk memastikannya terorganisir dengan baik dan efisien. Analisis persyaratan metodologi *waterfall*, desain, implementasi, pengujian, dan fase implementasi semuanya dijelaskan secara menyeluruh. Mengembangkan pemahaman yang kuat tentang persyaratan pengguna teknis dan memutuskan fungsionalitas yang harus dimiliki aplikasi adalah tugas yang terlibat dalam proses analisis kebutuhan. Antarmuka pengguna yang intuitif dan responsif kemudian diproduksi selama fase desain untuk meningkatkan pengalaman pengguna. Kata Perhatian utama dari artikel ini adalah bagaimana metodologi *waterfall* dapat membantu menyelesaikan masalah tantang saat mengembangkan aplikasi Android yang kompleks. Kami membahas banyak fase metode *waterfall* seperti analisis kebutuhan, desain, implementasi, pengujian, dan perencanaan, dan kami mengilustrasikan bagaimana setiap tahapan berkontribusi pada hasil akhir aplikasi *Technes*. Temuan diskusi menunjukkan bahwa metode *waterfall* dapat membantu mengatur, melacak, dan mengevaluasi setiap tahap pengembangan aplikasi dengan cara yang jelas dan terorganisir. Kami juga menyadari keterbatasan metode *waterfall* sehubungan dengan perubahan kebutuhan pengguna dan operasi *backend*, namun masalah ini dapat diselesaikan dengan komunikasi yang efektif. Hasil akhirnya adalah aplikasi *Technes* yang sepenuhnya memenuhi kebutuhan pengguna, memiliki agen pengguna yang responsif, dan menawarkan pengalaman pengguna yang konsisten.

**Kata Kunci:** Aplikasi Mobile, Android, E-Commerce, Firebase, Metode Waterfall

### **ABSTRACT**

This article describes how the *waterfall* method approach was used for the creation of an Android app by *Technes*, an app-based service provider. The *waterfall* method is a *methodical* and *linear* approach to developing software that starts with analyzing requirements and progresses through design, testing, and implementation. In this article, we will discuss how the *waterfall* method was applied to the development of the *Technes* app to ensure it was well-organized and efficient. The *waterfall* methodology's requirements analysis, design, implementation, testing, and implementation phases are all thoroughly explained. Developing a solid understanding of the technical user requirements and deciding on the functionality that the app should have are the tasks involved in the requirements analysis process. An intuitive and responsive user interface is then produced during the design phase to enhance the user experience. The main concern of this article is how the *waterfall* methodology can help solve challenging problems when developing complex Android



*applications. We discuss many phases of the waterfall method such as requirements analysis, design, implementation, testing, and planning, and we illustrate how each stage contributes to the final outcome of the Technes application. The findings of the discussion show that the waterfall method can help organize, track, and evaluate each stage of app development in a clear and organized way. We also recognize the limitations of the waterfall method with respect to changing user requirements and backend operations, but these issues can be resolved with effective communication. The end result is a Technes application that fully meets user needs, has a responsive user agent, and offers a consistent user experience.*

**Keywords:** Mobile Application, Android, E-Commerce, Firebase, Waterfall Model

*This is an open access article under the [CC BY-SA](#) license*



## 1. PENDAHULUAN

Penerapan Teknologi Informasi (TI) saat ini sangatlah berpengaruh di dunia kerja dalam mengambil keputusan. Kebutuhan akan informasi yang cepat, tepat, dan akurat sangat penting [1]. Tidak dapat disangkal bahwa perkembangan kemajuan teknologi di sekitar kita, khususnya teknologi informasi yang membuat data dan informasi menjadi mudah diakses konsumennya, telah bermanfaat bagi kemajuan manusia. Karena persyaratan untuk akses di hampir setiap perusahaan, organisasi, dan institusi, sangat didukung untuk proses kerja yang mulus. Data dan informasi yang tepat waktu dan efektif digunakan untuk meningkatkan layanan pelanggan [2]. Kebutuhan akan layanan yang cepat, efektif, dan mudah didapat semakin berkembang di era digital saat ini. Sektor jasa adalah salah satu yang sangat dipengaruhi oleh kemajuan teknologi. Konsumen semakin menuntut jawaban yang efisien dan dapat diandalkan untuk masalah mereka, terutama pada pemeliharaan dan perbaikan peralatan elektronik, peralatan rumah tangga, atau kendaraan, baik kendaraan roda dua maupun roda empat.

Penggunaan aplikasi dalam membantu pekerjaan manusia dinilai dapat meningkatkan kecepatan dan lebih efisien dari segi tenaga, waktu ataupun biaya tanpa menurunkan kualitas hasil yang dikerjakan [3]. Masyarakat dapat memanfaatkan ponsel pintar untuk semua kebutuhan yang memudahkannya termasuk dapat melakukan transaksi secara online dengan menggunakan aplikasi *e-commerce*. Aplikasi *e-commerce* merupakan aplikasi transaksi dalam bentuk digital. Aplikasi *e-commerce* dapat membantu *customer* dan *produsen* dalam melakukan transaksi jual beli barang atau jasa dengan lebih cepat dan mudah [4]. Pemanfaatan teknologi smartphone pada berbagai aspek kini telah dirasakan dampaknya, tanpa terkecuali sampai pada usaha jasa layanan *service*. Manusia pada umumnya menginginkan segala sesuatu dapat dengan mudah dikerjakan, begitu pula dengan layanan *service* yang dapat diakses dan dipesan kapan saja dan dari mana saja.

Ada beberapa penelitian yang dipandang relevan dengan penelitian ini, yaitu diantaranya penelitian yang dilakukan oleh Janiver W. dengan judul Rancang Bangun Aplikasi Online Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi [5]. Hasil penelitian ini menunjukkan bahwa perancangan dan pembangunan aplikasi online sistem pemesanan jasa tukang bangunan berbasis lokasi dapat memudahkan masyarakat dalam mencari jasa tukang bangunan yang ada di Kota Manado.

Selain itu, penelitian yang dilakukan oleh Nurhasan Nugroho dengan judul Pengembangan Aplikasi Pencarian Guru Privat Editing Video Berbasis Android [6], memperlihatkan bahwa sistem dapat digunakan dengan mudah dengan hasil rata-rata 82,5% responden menyatakan “setuju”, dan masuk dalam kategori baik. Akan tetapi aplikasi yang dikembangkan masih banyak kekurangan Diantaranya adalah proses transaksi dan penjadwalan masih dilakukan diluar sistem.

Berdasarkan penelitian sebelumnya, *Technes* menjadi pilihan yang sangat umum dalam situasi ini. Pelanggan dapat dengan cepat mengidentifikasi, membeli, dan membayar layanan - layanan profesional menggunakan aplikasi jenis ini tanpa harus membuang waktu mencari informasi atau menghubungi penyedia layanan. Oleh karena itu, aplikasi *Technes* telah mengubah cara klien menerima dukungan teknis dan pemeliharaan, yang menguntungkan kedua belah pihak. Selain itu, aplikasi ini membantu industri jasa tumbuh secara ekonomi, memberi para ahli dan bisnis akses ke pasar yang lebih besar, dan mendukung penciptaan ekosistem digital kreatif.

Dengan menawarkan layanan pelanggan sebesar mungkin dan akuntabilitas lengkap saat menangani keluhan, layanan harus mampu bersaing. Karena memberikan tingkat layanan berkualitas tinggi dan menyenangkan memiliki dampak signifikan pada peringkat pelanggan, Pelanggan juga akan pergi jika apa yang mereka dapatkan tidak mencukupi.

## **2. METODE PENELITIAN**

### **2.1 Metode Perancangan Sistem**

Teknik Siklus Hidup Pengembangan Sistem digunakan dalam pengembangan aplikasi *Technes*. *Waterfall* merupakan salah satu pendekatan SDCL yang sering digunakan. Secara konseptual. Pendekatan ini merupakan pendekatan berurutan untuk pengembangan perangkat lunak yang dimulai dengan perencanaan ide dan bergerak melalui pemodelan (desain), implementasi, pengujian, dan pemeliharaan. Analisis sistem secara keseluruhan sering menggunakan model *waterfall* ( Teknik Air Terjun). Ide mendasar di balik teknik air terjun adalah bahwa pengembangan sistem dilakukan secara berurutan atau linier. Oleh karena itu, langkah 2 tidak dapat diselesaikan jika langkah 1 belum selesai, dalam hal langkah kedua harus diselesaikan sebelum melanjutkan ke langkah berikutnya, dan seterusnya. Langkah 3 akan selesai secara otomatis jika langkah 1 dan 2 sudah selesai [14].

Tahapan pembuatan model *sekuensial linier* atau model pengembangan *waterfall* adalah sebagai berikut [15][16]:

a. *System Engineering*

Langkah pertama dimulai dengan membangun seluruh elemen sistem dan memilih bagian mana yang akan digunakan sebagai bahan pengembangan perangkat lunak, dengan mempertimbangkan hubungan dengan Perangkat Keras, Pengguna dan Basis Data.

b. *Analysis*

Analisis persyaratan perangkat lunak Mempelajari dan menyusun persyaratan sistem, termasuk untuk domain informasi, fungsi terkait kinerja/kinerja, dan antarmuka, adalah bagian dari proses ini. Temuan yang terdokumentasi dari analisis dan pengumpulan akan diteruskan ke pelanggan.

c. *Design*

Persyaratan untuk desain perangkat lunak diterjemahkan pada fase desain sehingga dapat diantisipasi sebelum pengkodean dilakukan. Struktur data, arsitektur perangkat lunak, representasi antarmuka, dan spesifikasi algoritma prosedural adalah penekanan dari prosedur ini.

d. *Code*

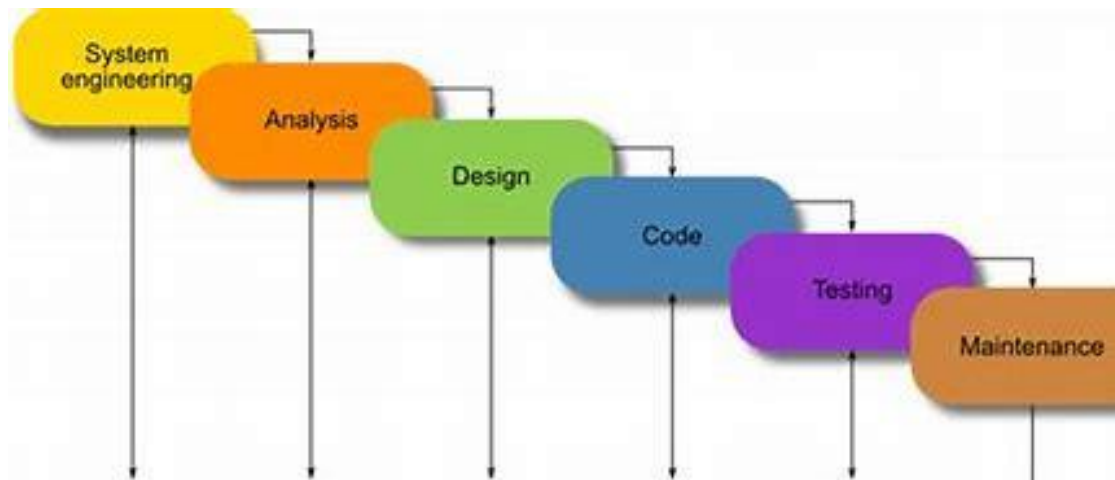
Menggunakan bahasa pemrograman, pengkodean melibatkan perubahan desain-desain menjadi bahasa yang dapat dipahami komputer.

e. *Testing*

Program perangkat lunak diuji, menggunakan pengujian logika internal dan pengujian eksternal fungsional untuk mencari potensi masalah dan menentukan apakah hasil pengembangan sesuai dengan yang dimaksudkan.

f. *Maintenance*

Langkah terakhir dalam siklus pengembangan adalah prosedur pemeliharaan, yang dilakukan setelah perangkat lunak digunakan.

Gambar 1. Metode *Waterfall*

## 2.2 Bahan dan Spesifikasi Kebutuhan Aplikasi

Untuk membuat aplikasi ini diperlukan banyak perangkat pendukung, baik dari segi *hardware* maupun *software*. Perangkat keras dan perangkat lunak yang digunakan dijelaskan secara lebih rinci di bawah ini:

a. Laptop & Spesifikasi :

- Processor : Intel Core i3 6006U
- Versi : Windows 10
- RAM : 4GB DDR4
- Harddisk : 512GB

b. *Software* yang digunakan :

- Android Studio
- Firebase
- Figma

c. *Smartphone* :

- Processor : Snapdragon 712
- OS : Android 11
- RAM : 4GB
- ROM : 128GB



```
imageButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent i = new Intent(getApplicationContext(), Otomotif.class);  
        startActivity(i);  
    }  
});  
  
konfirmasi1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String getName = nama_pelanggan.getText().toString();  
        String getJenis = jenis_layanan_servis.getText().toString();  
        String getLayanan = layanan_servis.getText().toString();  
        String getAlamat = alamat_pemesanan.getText().toString();  
  
        database.child("Pelanggan").push().setValue(new Customer(getName,  
getLayanan, getJenis, getAlamat));  
        Toast.makeText(Order_SepedaMotor.this, "Berhasil dikonfirmasi",  
Toast.LENGTH_SHORT).show();  
        startActivity(new Intent(Order_SepedaMotor.this, Dashboard.class));  
        finish();  
    }  
});  
}
```

Gambar 2. Source Code Aplikasi

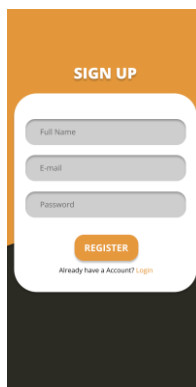
Gambar 2 merupakan kode android java untuk kelas program Order\_SepedaMotor. Kode tersebut digunakan untuk menyimpan data yang dimasukkan oleh pengguna di database *Firestore*. Kelas Order\_SepedaMotor adalah kelas yang mengimplementasikan metode *AppCompatActivity* yang berisi beberapa *variable* seperti *DatabaseReference*, nama, jenis layanan, layanan, alamat, dan tombol konfirmasi 1. Metode *setOnClickListener* ditambahkan pada *variable* tombol konfirmasi 1 untuk mengirim data ke *database child Firestore* dengan nama direktori "Pelanggan".

### 3.1 Halaman *Splash Screen*

Gambar 3. *Splash Screen*

*Splash Screen* adalah halaman yang menampilkan gambar logo atau animasi yang muncul pada awal pembukaan aplikasi selama beberapa detik sebelum aplikasi benar-benar dimuat. *Splash screen* biasanya dibuat dengan tujuan untuk memberikan kesan yang baik bagi pengguna atau memberikan identitas pada aplikasi tersebut sebelum pengguna mulai menggunakan aplikasi. *Splash screen* biasanya juga digunakan untuk memberi tahu pengguna jika aplikasi sedang memuat atau mempersiapkan diri sebelum benar-benar dijalankan.

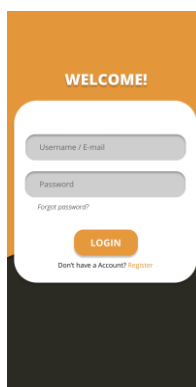
### 3.2 Halaman Registrasi



Gambar 4. Halaman Registrasi

Selanjutnya adalah halaman registrasi, yang merupakan halaman yang digunakan pada sebuah aplikasi untuk memungkinkan *user* untuk mendaftar atau membuat akun baru. Halaman registrasi seringkali terdiri dari *form* yang meminta *user* untuk mengisi informasi seperti nama, alamat *email*, dan *password*. Setelah *form* diisi dan dikirim, aplikasi akan memproses dan menyimpan informasi ke dalam database sehingga *user* dapat melakukan login dan menggunakan fitur-fitur pada aplikasi.

### 3.3 Halaman Login

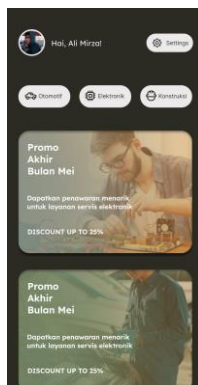


Gambar 5. Halaman Login

Halaman *login* adalah halaman yang digunakan untuk memungkinkan *user* untuk masuk ke dalam akun yang sudah terdaftar sebelumnya pada halaman register. Halaman *login* biasanya terdiri dari *username* atau *email* dan *password* untuk mengidentifikasi akun *user*. Setelah *user* memasukkan *username* atau *email* dan *password*, aplikasi akan memverifikasi informasi tersebut dengan data yang tersimpan dalam database. Jika data yang dimasukkan benar, maka *user* akan diarahkan ke halaman beranda atau *dashboard* di dalam aplikasi atau tersebut. Namun, jika data yang dimasukkan salah, maka *user* akan diminta untuk memasukkan kembali *username* atau *email* dan *password* dengan benar.



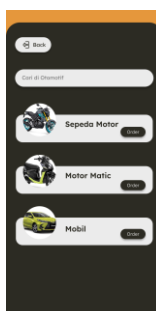
### 3.4 Halaman *Dashboard*



Gambar 6. Halaman *Dashboard*

Halaman *Dashboard* ini merupakan halaman utama yang menampilkan informasi mengenai fitur yang tersedia pada aplikasi serta beberapa konten atau informasi penting lainnya. Halaman ini memuat 3 jenis layanan *service* yang disediakan oleh aplikasi, yaitu layanan *service* otomotif, layanan *service* elektronik, dan layanan *service* konstruksi.

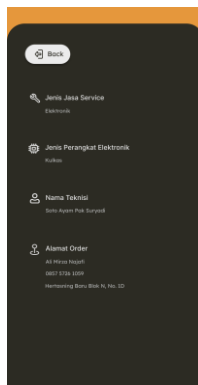
### 3.5 Halaman Layanan *Service*



Gambar 7. Halaman Jenis Layanan

Karena pada halaman utama memiliki 3 jenis layanan, pada halaman ini kita dapat memilih menu *order* pada jenis layanan otomotif. Halaman ini merupakan salah satu tampilan halaman yang akan memuat layanan-layanan *service* yang disediakan oleh aplikasi.

### 3.6 Halaman Detail Pemesanan

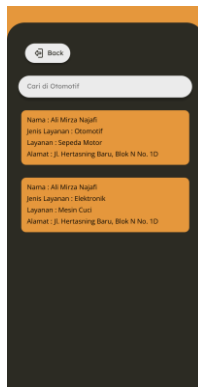


Gambar 8. Halaman Detail Pemesanan

Halaman Detail Pemesanan menampilkan informasi mengenai data pemesanan, seperti nama pelanggan, jenis layanan *service*, bentuk layanan *service*, dan alamat pelanggan. Data inilah yang nantinya akan dikirim dan disimpan ke *database* dan ditampilkan pada halaman riwayat pemesanan.



### 3.7 Halaman Riwayat Pemesanan



Gambar 9. Halaman Riwayat Pemesanan

Halaman Riwayat Pemesanan menampilkan informasi mengenai data riwayat pemesanan yang berisi nama pelanggan, jenis layanan *service*, bentuk layanan *service*, dan alamat pelanggan. Data inilah yang sebelumnya dikirim dan disimpan di database sebelum dipanggil pada halaman ini.

## 4. KESIMPULAN DAN SARAN

Metode *waterfall* menggunakan proses langkah demi langkah yang teratur dan *linier*, yang telah terbukti bermanfaat dalam membuat aplikasi *Technes*. Fase analisis persyaratan, desain, implementasi, pengujian, dan penyebaran memberikan tim pengembangan kerangka kerja yang jelas untuk mematuhi tugas yang diperlukan dalam menghasilkan aplikasi. Kami mengembangkan antarmuka pengguna yang responsif dan mudah digunakan selama fase desain. Tata letak ini efektif, navigasinya mudah digunakan, dan aspek estetikanya menarik. Aplikasi *Technes* yang dirancang dengan baik mudah digunakan dan meningkatkan pengalaman pengguna. Pada tahap implementasi, sekelompok pemrogram membuat aplikasi dengan fitur yang telah ditentukan menggunakan bahasa pemrograman Android dan kerangka kerja terkait. Kami memastikan kode yang kami hasilkan pada fase ini mematuhi standar tertinggi untuk pengembangan aplikasi Android.

Setelah implementasi selesai, aplikasi diuji dan divalidasi untuk memastikan berfungsi dengan baik dan memberikan pengalaman positif kepada pengguna. Prosedur pengujian menyeluruh membantu dalam menemukan dan memperbaiki setiap kesalahan atau kesalahan yang mungkin ada di aplikasi. Setelah memastikan bahwa aplikasi tersebut stabil dan memenuhi persyaratan kualitas yang ditetapkan, akhirnya kami melanjutkan ke proses implementasi. Kami juga menginstruksikan pelanggan dan teknisi tentang cara menggunakan aplikasi *Technes* dengan sukses. Meskipun metode *waterfall* memiliki kelebihan untuk mengatur pengembangan aplikasi, ada juga kekurangannya. Salah satu masalah utama dengan teknik air terjun adalah ketidakmampuannya untuk menyesuaikan perubahan kebutuhan yang diantisipasi selama pengembangan. Oleh karena itu, perencanaan menyeluruh dan pemahaman tentang permintaan pengguna sangat penting untuk mencegah perubahan besar di kemudian hari.

Secara keseluruhan, penggunaan metode *waterfall* dalam proses pengembangan aplikasi *Technes* Android telah membantu memastikan pengembangan yang terorganisir, efektif, dan sesuai dengan kebutuhan pengguna. Pendekatan ini menawarkan struktur yang ditentukan yang dapat digunakan oleh tim pengembangan untuk melaksanakan setiap tahap secara efektif. Metode *waterfall* digunakan dalam pembuatan aplikasi *Technes*, dan menawarkan manfaat termasuk perencanaan terorganisir, peningkatan efisiensi pengembangan, dan hasil yang dapat diprediksi. Tim pengembangan *Technes* dapat memastikan bahwa setiap langkah pengembangan selesai sebelum melanjutkan ke tingkat berikutnya dengan menggunakan metode *waterfall*. Akibatnya, kemungkinan kesalahan terjadi lebih rendah karena setiap tahap dapat diperiksa secara menyeluruh sebelum melanjutkan ke tahap berikutnya. Metode *waterfall* juga membantu dalam manajemen proyek yang lebih baik. Tim pengembangan mungkin memiliki kesadaran menyeluruh tentang kemajuan proyek, jadwal, dan alokasi sumber



daya yang dibutuhkan berkat tahapan yang jelas. Akibatnya, kontrol dan manajemen proyek menjadi lebih terorganisir dan efisien.

## REFERENSI

- [1] H. Kurniawan, W. Aprilia, I. Kurniawan, and D. Firmansyah, "Penerapan Metode *Waterfall* Dalam Perancangan Sistem Informasi Penggajian Pada SMK Bina Karya Karawang," *J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. Dan Komun.*, vol. 14, no. 4, pp. 13–23, Jan. 2020, doi: 10.35969/interkom.v14i4.58.
- [2] S. Sudirja, F. Faradillah, and H. A. Awanis, "Implementasi Model *Waterfall* Pada Sistem Informasi Akuntansi Penjualan Jasa Service Mobil," *J. Infortech*, vol. 2, no. 1, pp. 127–132, Jun. 2020, doi: 10.31294/infortech.v2i1.8287.
- [3] I. Ahmad, A. T. Prastowo, E. Suwarni, and R. I. Borman, "PENGEMBANGAN APLIKASI *ONLINE DELIVERY* SEBAGAI UPAYA UNTUK MEMBANTU PENINGKATAN PENDAPATAN".
- [4] A. P. Pratama and R. A. Zunaidi, "Implementasi scrum model dalam pengembangan aplikasi *e-commerce* pada bidang jasa pembangunan rumah," *JENIUS J. Terap. Tek. Ind.*, vol. 4, no. 1, pp. 39–48, May 2023, doi: 10.37373/jenius.v4i1.484.
- [5] Janiver W. Janis, Dringhuzen J. Mamahit, Brave A. Sugiarto, Arthur M. Rumagit, "Rancang Bangun Aplikasi *Online* Sistem Pemesanan Jasa Tukang Bangunan Berbasis Lokasi," 2020.
- [6] N. Nugroho, R. Napianto, I. Ahmad, W.A. Saputra, "Pengembangan Aplikasi Pencarian Guru Privat *Editing* Video Berbasis Android," 2021.
- [7] M. Sari and A. Asmendri, "Penelitian Kepustakaan (*Library Research*) dalam Penelitian Pendidikan IPA," *Nat. Sci.*, vol. 6, no. 1, pp. 41–53, Jun. 2020, doi: 10.15548/nsc.v6i1.1555.
- [8] H. Nur, "Penggunaan Metode *Waterfall* Dalam Rancang Bangun Sistem Informasi Penjualan," *Gener. J.*, vol. 3, no. 1, p. 1, Jan. 2019, doi: 10.29407/gj.v3i1.12642.
- [9] H. F. Harumy and H. M. Z. N. Amrul, "Aplikasi *Mobile* Zagiyan ( Zaringan Digital Nelayan) Dalam Menunjang Produktivitas Dan Keselamatan, Dan Kesehatan Nelayan ( Studi Kasus Kelompok Nelayan Percut)," *IT J. Res. Dev.*, vol. 2, no. 2, pp. 52–61, Mar. 2018, doi: 10.25299/itjrd.2018.vol2(2).1249.
- [10] R. Hermiati, A. Asnawati, and I. Kanedi, "PEMBUATAN E-COMMERCE PADA RAJA KOMPUTER MENGGUNAKAN BAHASA PEMROGRAMAN PHP DAN DATABASE MYSQL," *J. MEDIA INFOTAMA*, vol. 17, no. 1, Feb. 2021, doi: 10.37676/jmi.v17i1.1317.
- [11] F. Irvansyah, S. Setiawansyah, and M. Muhaqiqin, "APLIKASI PEMESANAN JASA CUKUR RAMBUT BERBASIS ANDROID," *J. Ilm. Infrastruktur Teknol. Inf.*, vol. 1, no. 1, pp. 26–32, Jun. 2020, doi: 10.33365/jiiti.v1i1.253.
- [12] S. Mulyati, "Kreativitas Matematis Siswa Pada Pembelajaran Discovery Learning Dengan Media Berbasis Android Studio," 2019.
- [13] N. Nursyifa, M. Ridwan, Arif Jaya, Andi Syarifuddin, BA Ashad, "Rancang Bangun Smart Home Berbasis Internet Of Things Menggunakan Firebase Real Time Database Dan Aplikasi Android," 2022.
- [14] A. Ardiansyah, D. Pratmanto, S. Aji, and F. Fandhilah, "Sistem Informasi Jasa Servis Printer Dengan Metode *Waterfall*," *Indones. J. Softw. Eng. IJSE*, vol. 8, no. 1, pp. 18–25, Jun. 2022, doi: 10.31294/ijse.v8i1.12747.



- [15] M. Riastuti and Y. I. Chandra, "Perancangan Aplikasi Pelayanan Service Bengkel Motor ABS Menggunakan Model Sequential Linier Berbasis Android," *J. Esensi Infokom J. Esensi Sist. Inf. Dan Sist. Komput.*, vol. 6, no. 1, pp. 64–71, May 2022, doi: 10.55886/infokom.v6i1.458.
- [16] Surianto, D. F. (2016). *Rancang Bangun Sistem Informasi X-CORE (Exploring the Campus Organizations Events) berbasis Android dengan menggunakan Ionic Framework* (Undergraduate Thesis, UNIVERSITAS BAKRIE).