

**PEMBAYARAN HONORARIUM EVALUATOR PERGURUAN TINGGI
KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
DENGAN APLIKASI HANOVA BERBASIS ANDROID
(studi kasus di bidang perencanaan anggaran kementristek DIKTI)**

Al Achtur Djamil¹, Fitri Latifah²,
Trknik Informatika¹, Teknik Informatika²
Sekolah Tinggi Informatika dan Komputer Nusa Mandiri¹,
Sekolah Tinggi Informatika dan Komputer Nusa Mandiri²
Email al.djamil19@gmail.com¹, fitri.flr@nusamandiri.ac.id²

Abstrak

Kemajuan Teknologi dan perkembangan teknologi di Indonesia menyebabkan semakin banyaknya pertumbuhan perguruan tinggi di Indonesia semakin cepat, dengan keadaan seperti ini menyebabkan semakin banyak terjadi pengusulan pendirian perguruan tinggi dan penambahan program studi, namun tentu saja pihak yang berwenang dalam hal ini ristekdikti tidak serta merta memberikan kemudahan dalam pengusulan perguruan tinggi baru dan penambahan program studi baru, setiap pengusulan perguruan tinggi dan penambahan program studi baru tentunya harus memenuhi syarat- syarat yang tidak mudah, serta akan ada penilaian kinerja dari perguruan tinggi pengusul, sengan semakin banyak nya pengusulan perguruan tinggi bau dan penambahan program studi baru mengakibatkan ristekdikti membutuhkan banyak evaluator untuk melakukan proses evaluasi pada setiap usulan yang diajukan pada Kementerian Riset, Teknologi dan Pendidikan Tinggi, tentu saja hal ini mengakibatkan ristekdikti membutuhkan layanan yang cepat untuk penyampaian informasi pembayaran evaluator. Aplikasi berbasis mobile yang akan dibuat peneliti memberi nama HANOVA yang memanfaatkan data dari Direktorat Pengembangan Kelembagaan Perguruan Tinggi. Aplikasi ini menggunakan REST *service* untuk memanipulasi data yang ada pada layanan web hosting. Dengan aplikasi mobile ini dapat mempercepat informasi pembayaran evaluator dari pihak Kementerian Riset, Teknologi dan Pendidikan Tinggi dengan evaluator.

Kata Kunci: Evaluator, Kementerian, REST

Abstract: Technological advancements and technological developments in Indonesia have led to an ever faster growth of higher education institutions in Indonesia, with conditions like this leading to more and more proposals for the establishment of educational institutions. higher education and the addition of study programs, but of course the authorities in this case Ristekdikti do not necessarily provide facilities by offering new higher education institutions and adding new study programs, each institution d higher education proposed and adding new study programs must of course meet the requirements that are not easy, and there will be an evaluation of the performance of higher education institutions, as more and more proposals for smelly higher educational institutions and additions to the new study program es led Ristekdikti to require many evaluators to carry out an evaluation process on each proposal submitted to the Ministry of Research, Technology and Higher Education, which of course led Risteikdikti to require prompt services for the payment information assessor. The mobile application which will be produced by the researchers gives the name of HANOVA which uses data from the Directorate of Institutional Development of Higher Education. This application uses the REST service to manipulate existing data on web hosting services. With this mobile app, you can speed up the assessor's payment information with the Ministry of Research, Technology and Higher Education with the assessor.

Keywords: Evaluator, Ministry, REST

JISAMAR (Journal of Information System, Applied, Management, Accounting and Research)

<http://journal.stmikjayakarta.ac.id/index.php/jisamar> Telp. +62-21-3905050

e-mail: jisamar@jayakarta.ac.id , jisamar2017@gmail.com

I. PENDAHULUAN

Saat ini pertumbuhan perguruan tinggi di Indonesia semakin cepat, menyebabkan pengusulan pendirian perguruan tinggi dan penambahan program studi bertambah, sehingga membutuhkan evaluator yang yang tidak sedikit dalam melakukan proses evaluasi pada setiap usulan yang diajukan pada Kementerian Riset, Teknologi dan Pendidikan Tinggi, pada setiap satu bulan admin keuangan harus melakukan pendataan terhadap setiap usulan yang telah selesai di evaluasi untuk melakukan proses pembayaran. Karena semakin banyak evaluator yang harus di bayar sehingga admin keuangan kesulitan dalam menginformasikan evaluasi yang sudah dilakukan pembayaran. android adalah sebuah operating system yang dikhususkan untuk perangkat mobile (bukan PC). Basis dari Android adalah open source yang dikenal dengan nama Linux. Sebelumnya Android Inc. berdiri secara independen yang didirikan pada tahun 2003 oleh Andy Rubin, Rich Miner, Chris White, dan Nicks Sears. Android Inc. bermarkas di Palo Alto, California. Perusahaan berusaha untuk mengembangkan sebuah operating system mobile yang lebih canggih dari operating system mobile kala itu. Kemudian pada Agustus 2005 Google mengajukan penawaran untuk mengakuisisi perusahaan tersebut. Sistem yang terdapat saat ini berbasis web, dimana tidak semua evaluator selalu update terhadap informasi pembayaran yang disampaikan melalui web, sehingga mengakibatkan banyak dari evaluator komplain karena tidak mendapatkan informasi pembayaran terhadap setiap evaluasi yang telah mereka lakukan. Oleh sebab itu penulis mencoba untuk membuat aplikasi berbasis android untuk perancangan sistem informasi pembayaran evaluator perguruan tinggi di Kementerian Riset, Teknologi dan Pendidikan Tinggi dengan menggunakan algoritma quick shorting. Dengan harapan dibuatnya aplikasi ini dapat mempermudah evaluator untuk selalu update atas informasi pembayaran yang disampaikan dan dapat melakukan pengecekan dengan mudah.

II. LITERATUR DAN METODE

Pada pembuatan aplikasi HANOVA ini peneliti merujuk kepada penelitian – penelitian sebelumnya baik dalam pembuatan aplikasi ataupun dalam konsep teoritis yang pernah di

lakukan oleh peneliti sebelumnya. Dalam perancangan aplikasi HANOVA ini peneliti menggunakan konsep Objek Oriented Program (OOP) merujuk dari penelitian Kadek Wibowo, bahwa pemrograman berorientasi objek atau *Object Oriented Programming* merupakan pendekatan yang menggunakan *object* dan *class* [1]. OOP bukan sekedar bagaimana cara menuliskan sintaks program akan tetapi lebih dari itu OOP adalah cara pandang pada saat menganalisa suatu sistem serta permasalahan pemrograman. Pada OOP untuk setiap coding (program) adalah objek dan setiap objek mewakili bagian dari suatu program yang akan diselesaikan, masih merujuk pada penelitian Kadek Wibowo, dalam pembuatan aplikasi HANOVA ini penulis menggunakan arsitek tur MVC (*Model View Controller*). Metode MVC telah banyak digunakan dalam aplikasi yang mendukung sistem salah satunya dalam perancangan dan implementasi perangkat lunak. MVC ini membagi aplikasi menjadi tiga bagian yaitu : (1). Bagian Model, (2) Bagian View dan (3). Bagian *Controller*. Untuk pembuatan model dari aplikasi HANOVA ini peneliti menggunakan UML (*Univefied Modeling Language*) merujuk dari Sutejo [2]. Dalam membuat coding aplikasi ini peneliti menggunakan Kotline yaitu bahasa pemrograman bertipe data statis yang merujuk dari penelitian Bastari [3]

III. METODE

a. Metode Pengumpulan data

Dalam pengumpulan data peneliti menggunakan metode sebagai berikut

1. Metode Wawancara

Peneliti melakukan tanya jawab dengan para evaluator dan admin aplikasi sebagai pengguna aplikasi untuk mengetahui fitur-fitur apa saja yang mereka butuhkan saat menggunakan aplikasi HANOVA ini

2. Metode Observasi

Penulis melakukan observasi terhadap evaluator dan admin aplikasi, serta peneliti melakukan perbandingan terhadap aplikasi yang serupa yang telah dibangun sebelumnya.

3. Metode Studi Pustaka

Pengambilan data dengan cara mempelajari, mengkaji dan memahami sumber-sumber data yang ada pada beberapa literatur seperti buku-buku dan jurnal-jurnal.

b. Metode Pengembangan Aplikasi

Dalam pengembangan aplikasi HANOVA ini peneliti menggunakan metode Waterfall yang merupakan salah satu model dari metode SDLC (*System Development Life Cycle*). Model Waterfall ini merupakan contoh sebuah proses yang bersifat paln-driven dimana semua aktivitas yang akan dilakukan harus direncanakan terlebih dahulu sebelum mengerjakan Sommerville [4].

Adapun tahapan pembuatan aplikasi HANOVA dengan metode Waterfall adalah sebagai berikut :

1. Analisa

Dalam tahap ini yang dilakukan adalah menganalisa kebutuhan aplikasi, yang dilakukan oleh peneliti dalam mengumpulkan data sebagai bahan untuk merancang aplikasi

2. Desain

Di proses desain merupakan proses multi langkah dan berfokus pada rancangan aplikasi yang meliputi perancangan algoritma, rancangan database, rancangan permodelan aplikasi dengan UML , Proses desain ini juga merupakan proses penterjemahan hasil analisa ke dalam representasi perangkat lunak

3. Code

Pada tahapan ini desain diterjemahkan kedalam aplikasi, untuk implementasi dari tahap code ini bergantung dari hasil desain pada tahapan sebelumnya

4. Test

Setelah pengkodean langkah selanjutnya adalah melakukan pengujian aplikasi yang dibuat, pengujian ini dilakukan untuk mengetahui kesesuaian output dengan kebutuhan yang telah dirancang pada tahap sebelumnya.

5. Support

Pada tahap ini dilakukan pemrosesan dari coding yang dibuat pada perangkat yang memiliki spesifikasi yang sesuai, baik perangkat keras maupun perangkat lunak

c. Algoritma yang digunakan

IV. HASIL DAN PEMBAHASAN

a. Analisa Kebutuhan

1. Analisa Kebutuhan Fungsional

Kebutuhan ini merupakan kebutuhan layanan untuk pemakai aplikasi HANOVA.

2. Analisa Kebutuhan Non Fungsional

Analisa kebutuhan ini dibagi menjadi analisa kebutuhan perangkat keras dan analisa kebutuhan perangkat lunak, adapun kebutuhan perangkat keras sebagai berikut :

- a). Prosesor Intel Core i5
- b). Intel Iris Graphics 540 1536 MB
- c). Memeory 8GB DDR3
- d). Display 13,3 inchi
- e). SSD 250, 69 GB

Sedangkan kebutuhan perangkat lunak

- a). Android Studio
- b) SDK (*Software Development Kit*)
- c). MySQL
- d) WebHosting

b. Desain

1. Desain Algoritma

```

override fun onCreate(savedInstanceState: Bundle?, savedInstanceState: Bundle?) {
    val view = LayoutInflater.from(parentContext).inflate(R.layout.template_list_perguruan_tinggi, parent, false)
    return ListViewAdapter(this, view)
}

override fun getItemCount(): Int {
    return listData.size
}

override fun onBindViewHolder(holder: ViewHolder, position: Int) {
    holder.bind(listData[position], position)
}

class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
    fun bind(data: DataPerguruanTinggi): ViewHolder {
        with(itemView) {
            textViewPerguruan_tinggi.text = data.perguruan_tinggi
            textViewProgram_studi.text = data.program_studi
            itemView.setOnClickListener {
                val bundle = Bundle()
                bundle.putString("nama", data.nama)
                bundle.putString("pt", data.perguruan_tinggi)
                bundle.putString("ps", data.program_studi)
                bundle.putString("tgl_bayar", data.tgl_bayar)
                bundle.putString("jumlah", data.jumlah)
                bundle.putString("bank", data.bank)
                bundle.putString("sta", data.status_pembayaran)
                bundle.putString("jen", data.jenjang)
                bundle.putString("tgl_penugasan", data.tgl_penugasan)
                bundle.putString("tgl_selesai_penugasan", data.tgl_selesai_penugasan)
                val intent = Intent(this.context, DetailStatusActivity::class.java)
                intent.putExtras(bundle)
                context.startActivity(intent)
            }
        }
    }
}
    
```

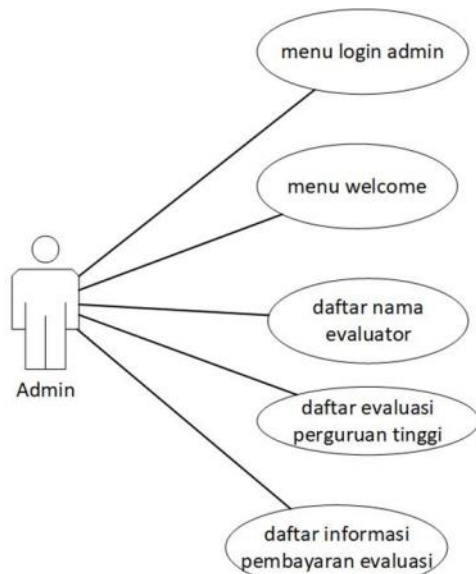
Gambar 4.1. Code Quick Sort

Sumber : Hasil penelitian Al Achtur, Fitri 2019

Algoritma yang digunakan dalam pembuatan aplikasi HANOVA ini adalah Quick Sort, pada saat menekan tombol informasi pembayaran evaluator makan akan diarahkan ke menu daftar nama evaluator dan disana list nama evaluator telah disorting berdasarkan abjad.

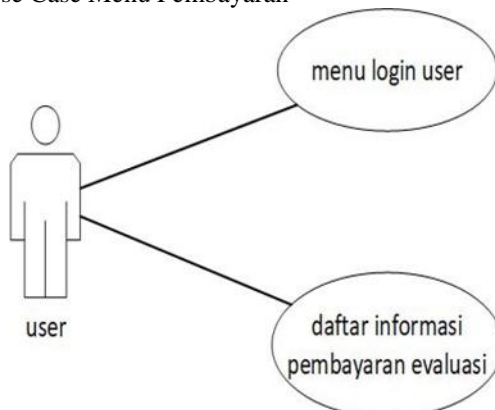
2. Desain Permodelan Aplikasi

a. Usecase Aplikasi HANOVA



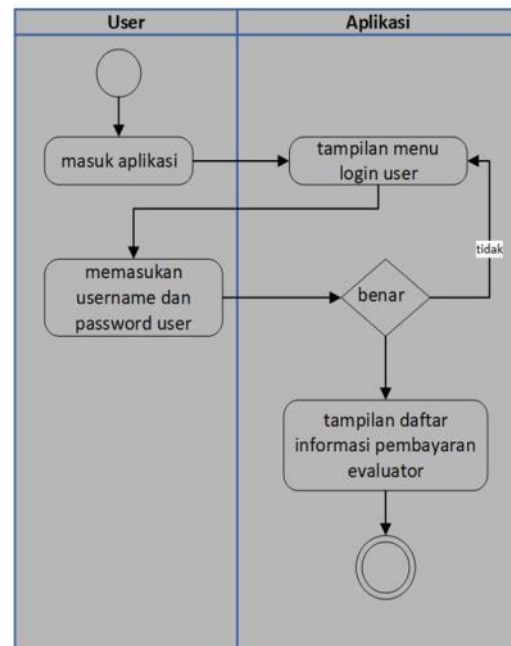
Gambar 4.2. Use Case Aplikasi HANOVA
Sumber : Hasil penelitian Al Achur, Fitri 2019

b. Use Case Menu Pembayaran



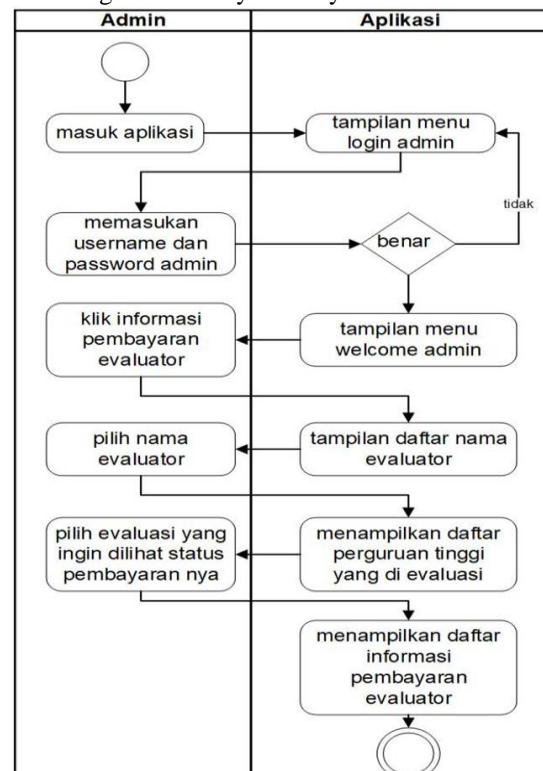
Gambar 4.3. Use Case Menu Pembayaran
Sumber : Hasil penelitian Al Achur, Fitri 2019

c. Diagram Activity Modul Pembayaran



Gambar 4.4. Diagram Activity Login
Sumber : Hasil penelitian Al Achur, Fitri 2019

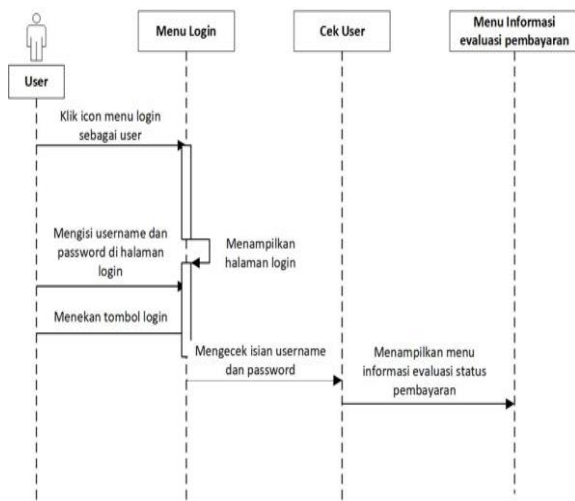
d. Diagram Activity Pembayaran evaluator



Gambar 4.5. Activity Pembayaran Evaluator

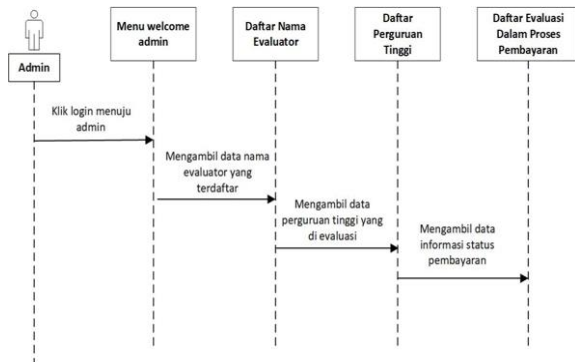
Sumber : Hasil penelitian Al Athur, Fitri 2019

e. Diagram Sequence User Aplikasi HANOVA



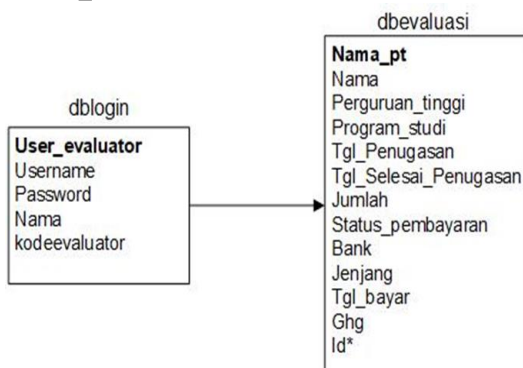
Gambar 4.6. Sequence User Aplikasi HANOVA
Sumber : Hasil penelitian Al Achtur, Fitri 2019

f. Diagram Sequence Admin Pembayaran Evaluator



Gambar 4.7. Sequence Admin Pembayaran Evaluator
Sumber : Hasil penelitian Al Achtur, Fitri 2019

3. E_RD



Gambar 4.8. E_RD Aplikasi HANOVA
Sumber : Hasil penelitian Al Achtur, Fitri 2019

c. Coding

1. Coding untuk login user pada aplikasi HANOVA

```

1 <?php
2 /* isi dengan koneksi data base*/
3 include("config.php");
4 if(isset($_GET['username']) && isset($_GET['password'])) {
5     $result = array();
6     $username = $_GET['username'];
7     $password = $_GET['password'];
8     $query = mysqli_query($conn, "SELECT * FROM 'user_evaluator' WHERE username='$username' AND password='$password'");
9     while ($row = mysqli_fetch_assoc($query)) {
10        array_push($result, array(
11            'nama' => $row['nama'],
12            'kodeevaluator' => $row['kodeevaluator']
13        ));
14    }
15    if($result){
16        echo json_encode(array(
17            'status' => 'sukses',
18            'result' => $result
19        ));
20    }else{
21        echo json_encode(array(
22            'status' => 'gagal',
23            'result' => $result
24        ));
25    }

```

Gambar 4.9. Login User Aplikasi HANOVA
Sumber : Hasil penelitian Al Achtur, Fitri 2019

d. Tesing

Pada pengujian aplikasi HANOVA ini peneliti menggunakan pengujian *black testing*, berikut tes yang telah dilakukan adalah sebagai berikut

Tabel 4.1. Hasil Tes Aplikasi HANOVA

| Halaman | Skenario | Hasil |
|-------------------------------------------|-------------------------------------------------------------------------|----------|
| Menu Login | Berhasil login sebagai admin dan user | Berhasil |
| Menu Nama Evaluator | Dapat menampilkan nama evaluator yang terdaftar dan klik nama evaluator | Berhasil |
| Menu Daftar Perguruan Tinggi | Dapat menampilkan data perguruan tinggi dan nama prodi yang di evaluasi | Berhasil |
| Menu Daftar Informasi Pembayaran Evaluasi | Dapat menampilkan rincian proses pembayaran evaluasi | Berhasil |

e. Support

Apliksi HANOVA ini dapat dijalankan pada smartphone android dengan minimal spesifikasi bagai berikut:

Tabel 4.2 Minimum Spesifikasi Aplikasi HANOVA

| Kebutuhan | Keterangan |
|---------------------|-------------------------------------------------|
| <i>SistemOprasi</i> | Android versi 4.0 atau <i>IceCream Sandwich</i> |
| Ram | 512 |

f. Implementasi Aplikasi HANOVA



Gambar 4.10 Login Pengguna
Sumber : Hasil penelitian Al Achtur, Fitri 2019



Gambar 4.11. Daftar Perguruan Tinggi
Sumber : Hasil penelitian Al Achtur, Fitri 2019

V. KESIMPULAN

Aplikasi pembayaran honorarium yang diberi nama HANOVA berhasil dibangun dengan menggunakan pemrograman untuk eclips dan SDK sebagai development. Aplikasi HANOVA ini diharapkan dapat membantu evaluator maupun bagian perencanaan anggaran ristek dikti dalam memberikan informasi evaluator tentang honorarium yang telah dibayarkan., sehingga pihak Kemenristekdikti tidak dan evaluator tidak lagi menanyakan tentang honorarium. Aplikasi ini belum sepenuhnya di lounching, mengingat masih banyak kekurangan-kekurangan yang ada sehingga perlu untuk disempurnakan oleh peneliti selanjutnya, sehingga pada akhirnya aplikasi ini benar-benar siap untuk di pergunakan.

REFERENASI

[1] Kadek Wibowo (2015). Analisa Konsep Object Oriented Programming pada Bahasa

Pemrograman PHP, 151-153

- [2] Sutejo (2016). Pemodelan UML Sistem Informasi Geografis Pasar Tradisional Kota Pekanbaru. *Jurnal Teknologi Informasi & Komunikasi Digital Zone*, Volume7, Nomor 2, November 2016.
- [3] Bastari, D. I., Pradana, F., & Priyambadha, B. (2017). Pengembangan Sistem Pembelajaran Pemrograman Java yang Atraktif Berbasis Website, 1(12),
- [4] Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus: Aplikasi Prediksi Kelulusan SNMPTN)*, I(3), 34. <https://doi.org/ISSN : 2407 - 3911>
- [5] Ade Djohar Maturidi. (2014) *Metode Penelitian Teknik Informatika*, (65)
- [6] Feridi. (2019). *Mengenal RWSRful Web Services - CodePolitan*
- [7] Kadek Wibowo (2015). *Analisa Konsep Object Oriented Programming pada Bahasa Pemrograman PHP*, 151-153
- [8] Anisya Sonita dan Febrina Nurtaneo (2017). *Analisis Perbandingan Algoritma Bubble Sort, Merge Sort dan Quick Sort dalam Proses Pengurutan Kombinasi Angka dan Huruf*, 77.
- [9] Vol, J., No, X. I. V, Staf, Y., & Upri, P. (2015). *Sistem Informasi Perpustakaan Berbasis Web*, XIV(2), 66–77.
- [10] Wijaya, B. D., Fenty, E. M. A., & Fiade, A. (2015). *Implementasi JSON PARSING pada Aplikasi Mobile E-Commerce*. *Jurnal Pseudocode*, 2(1), 1–9