

Perancangan *Web Service* Berbasis REST API Untuk Aplikasi Penerimaan Peserta Didik Baru

Ryan Afrizal¹, Fitriyani²

¹Universitas Adhirajasa Reswara Sanjaya
Jalan Sekolah Internasional 1-2 Antapani Bandung, 022-7100124
e-mail: ryan.afrizal20@gmail.com

²Universitas Adhirajasa Reswara Sanjaya
Jalan Sekolah Internasional 1-2 Antapani Bandung, 022-7100124
e-mail: fitriyani@ars.ac.id

Abstrak

Pada setiap tahun ajaran baru, di beberapa sekolah di Indonesia pasti akan terdapat antrean dari para calon siswa ataupun orang tua siswa yang ingin mendaftar atau mendaftarkan anaknya ke sekolah tersebut, antrean tersebut biasanya disebabkan oleh kurangnya persiapan dari pihak sekolah seperti kurangnya formulir pendaftaran. Melihat Permasalahan ini maka dibutuhkan sebuah sistem penerimaan peserta didik baru untuk membantu para calon siswa dalam proses pendaftaran agar tidak terjadi antrean di sekolah tersebut. Tujuan dari penelitian ini adalah membuat sebuah sistem aplikasi berbasis web dan mobile untuk membantu calon siswa atau orang tua dari calon siswa dalam proses pendaftaran peserta didik baru. Perancangan sistem ini menggunakan model pengembangan software dengan metode waterfall dan membuat sebuah REST API dengan menggunakan bahasa pemrograman Python dibantu dengan toolkit Django Rest Framework, dengan menggunakan REST API pada aplikasi penerimaan peserta didik baru yang dibuat sehingga aplikasi berbasis web dengan aplikasi berbasis mobile dapat terhubung satu sama lain.

Kata Kunci : Pendaftaran siswa, web service, REST API, web, mobile

Abstract

Every new school year there will certainly be a queue of prospective students or parents at different schools in Indonesia who want to enroll or enroll their children at the school, the queue is usually caused by a lack of preparation from the school such as a lack of application forms. Given this problem, a new student admission system is needed to help prospective students with the registration process so that there is no more queue at the school. The goal of this research is to create a web-based and mobile application system to assist prospective students in the enrollment process for new students. The design of this system uses a software development model with the waterfall method and creates a REST API using the Python programming language, supported by the Django Rest Framework Toolkit, so that new student enrollment applications created between web-based applications and mobile applications can be connected to each other.

Keywords: Student Admission, web service, REST API, web, mobile

1. Pendahuluan

Pendaftaran dan penerimaan peserta didik baru merupakan salah satu kegiatan yang diselenggarakan setiap tahun ajaran baru di setiap SMA (Sekolah Menengah Atas)

di Indonesia baik sekolah negeri maupun swasta. Pendaftaran dan penerimaan peserta didik baru adalah salah satu kegiatan manajemen peserta didik paling penting, karena dengan masuknya peserta didik baru

pada sebuah sekolah akan membawa dampak positif bagi sekolah tersebut sehingga kegiatan belajar mengajar dapat terus berjalan dengan maksimal (Sari dkk, 2016).

Hampir setiap tahun ajaran baru di setiap sekolah menengah atas di Indonesia melakukan penerimaan dan pendaftaran peserta didik baru, bahkan di beberapa sekolah terdapat antrian yang sangat panjang, antrian panjang tersebut disebabkan oleh banyaknya calon siswa yang ingin masuk ke sekolah tersebut tetapi tidak diiringi dengan persiapan yang matang dari pihak sekolah seperti: ketersediaan formulir pendaftaran dan sedikitnya petugas yang tersedia untuk membantu calon siswa dalam proses pendaftaran, belum lagi pada saat pengumuman penerimaan siswa, akan terjadi penumpukan siswa yang ingin memeriksa apakah dia diterima atau tidak di sekolah pilihannya. Seiring dengan berkembang pesatnya teknologi, internet sudah menjadi bagian dari kehidupan masyarakat, hal ini terbukti dengan banyaknya kegiatan manusia yang awalnya dilakukan secara manual sekarang bisa dilakukan dengan memanfaatkan fasilitas internet, seperti: pembelian barang, pemesanan transportasi dan masih banyak lagi. Begitu pula dengan pendaftaran peserta didik baru, di beberapa SMA di Indonesia sudah mulai menerapkan sistem pendaftaran peserta didik baru secara online (Kiki, 2020).

Perancangan sistem pendaftaran peserta didik baru secara *online* dapat dibuat dengan berbasis *web* maupun aplikasi *mobile*, tetapi jika ingin merancang sebuah sistem aplikasi berbasis *web* beserta aplikasi *mobile* akan memerlukan waktu yang lama dikarenakan harus merancang 2 *frontend* dan 2 *backend* yang berbeda, tetapi dengan memanfaatkan salah satu teknologi *web service* yaitu REST, dapat mempersingkat waktu dalam perancangan sistem aplikasi dikarenakan cukup membuat 1 *backend* yang dapat saling terhubung antara aplikasi berbasis *web* dengan aplikasi *mobile*. REST (*Representational State Transfer*) merupakan salah satu jenis dari *web service* yang memungkinkan *system request* dapat mengakses dan memanipulasi teks yang direpresentasikan oleh *web service*, REST terbukti lebih cepat dibandingkan dengan

metode *web service* yang serupa (Rulloh dkk, 2017:85).

Berdasarkan penjelasan yang telah dipaparkan sebelumnya, penulis bermaksud untuk membuat sebuah sistem aplikasi pendaftaran peserta didik baru dengan memanfaatkan fasilitas internet yang bertujuan untuk mempermudah para calon siswa untuk mendaftar dan dapat mengetahui informasi diterima atau tidaknya di sekolah tersebut.

Web Service

Menurut Ardhana dalam (Angga Diputra dkk, 2018) menyatakan bahwa “web adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan *surfer* (sebutan para pemakai komputer yang melakukan *browsing* atau penelusuran informasi melalui internet)”. Sedangkan web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan, web service yang umumnya dikirim melalui HTTP (*Hypertext Transfer Protocol*) (Sibagariang, 2016:29).

A. Jenis-Jenis Web Service

Menurut (Sibagariang, 2016:29) *web service* dibagi menjadi 2 jenis yaitu :

1. *Representational State Transfer* (REST)
2. *Simple Object Access Protocol* (SOAP)

REST (Representational State Transfer) API

REST (*Representational State Transfer*) merupakan salah satu jenis arsitektur untuk penerapan *web service* yang menerapkan konsep perpindahan antar *state*, setiap *state* mewakili satu *resource* yang ada pada server (Sy & Intan, 2019 :64). Sedangkan REST API atau *RESTful* API merupakan salah satu model implementasi dari *web service*.

REST adalah gaya arsitektural yang memiliki aturan seperti antarmuka yang seragam, sehingga jika aturan tersebut diterapkan pada *web service* akan dapat memaksimalkan kinerja *web service* terutama pada performa, skalabilitas dan kemudahan untuk dimodifikasi. Pada arsitektur REST, REST *server* menyediakan *resources* (sumberdaya) dan REST *client* mengakses dan menampilkan *resource* tersebut untuk

penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (*Universal Resource Identifiers*) atau global ID. *Resource* tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Representasi REST API dalam bentuk notasi JSON dan XML, dimana data-data yang terlihat dalam bentuk suatu objek atau *array*.

Python

Python merupakan bahasa pemrograman yang dikategorikan sebagai *highlevel language*. Berbeda dengan *lowlevel language*, *highlevel language* tidak dapat langsung dijalankan oleh mesin, perlu diproses terlebih dahulu agar dapat dijalankan oleh mesin (NATALIANA dkk, 2014:74).

Django

Django adalah sebuah *framework full-stack* untuk membuat aplikasi berbasis *web* dengan menggunakan bahasa pemrograman *python*, *django* dapat meliputi sisi *front-end* dan juga *back-end*. *Front-end* adalah sisi depan yang akan dilihat oleh pengguna, sedangkan *back-end* adalah sisi belakang yang berhubungan dengan *database* dan logika bisnis. *Framework* berfungsi untuk membantu dalam pembuatan sebuah *web* agar pembuatan lebih cepat (Muhardian, 2016).

Django *web framework* menggunakan konsep model MVT (*Model-View-Template*). Perbedaan konsep model MVT (*Model-View-Template*) yang digunakan dalam *web framework* *django* dibandingkan dengan konsep model MVC (*Model-View-Controller*) yang sering digunakan pada *web framework* lain seperti *laravel* dan *CI (codeignitier)* adalah jika dalam konsep MVC kode yang mengontrol antara *model* dan *view* adalah sebuah *controller* jika dalam konsep MVT kode yang mengontrol antara *modal* dan *view* adalah *template*. *Template* disini merupakan sebuah file HTML yang digabungkan dengan *Django Template Language (DTL)* (Kapalomen, 2016).

2. Metode Pengembangan Sistem

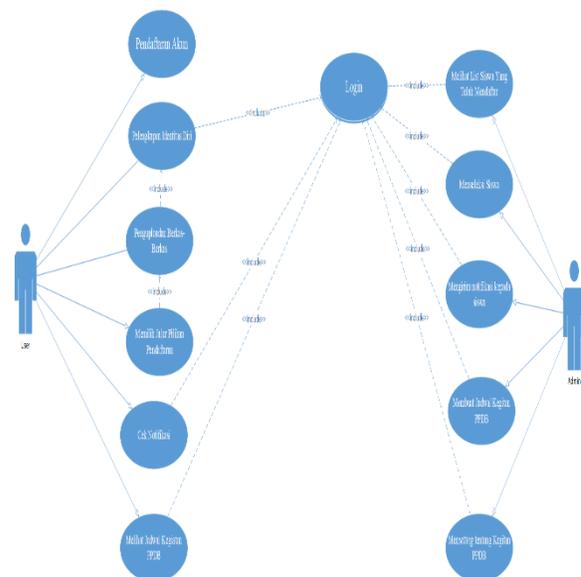
Metode pengembangan sistem atau *System Development Life Cycle (SDLC)* adalah metodologi umum yang digunakan dalam pengembangan suatu sistem informasi,

SDLC terdiri dari beberapa fase yang dimulai dari fase perancangan, analisis, perancangan, implementasi hingga pemeliharaan sistem (Susanto & Andriana, 2016:41).

Waterfall adalah salah satu dari beberapa metode yang digunakan dalam pengembangan sistem. Menurut Rosa & Shalahuddin dalam (Hadinata & Fitriyani, 2017:40) menyatakan bahwa "*waterfall* sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*), metode ini digunakan karena tahap-tahap yang dilakukan berurut sehingga memudahkan untuk menentukan waktu pengerjaan dari setiap tahapnya".

3. Hasil dan Pembahasan Use Case

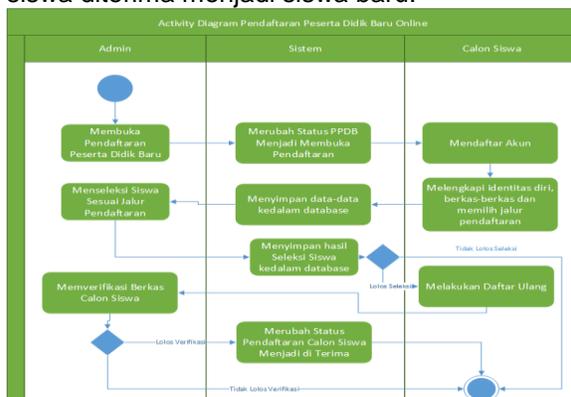
Use case diagram untuk aplikasi PPDB *online* ini memiliki 2 *actor* yaitu calon siswa dan *admin*. Untuk calon siswa dapat mendaftar akun, melengkapi identitas, melengkapi berkas-berkas, memilih jalur pendaftaran, cek notifikasi, melihat jadwal kegiatan PPDB, sedangkan untuk *admin* dapat melihat list siswa yang telah mendaftar, menseleksi siswa, mengirim notifikasi kepada siswa, membuat kegiatan PPDB, merubah status pendaftaran PPDB. Berikut ini merupakan *use case* diagram dari aplikasi Pendaftaran Peserta Didik Baru (PPDB) *Online*, dapat dilihat pada Gambar III.1.



Gambar 1. Use case Aplikasi PPDB

Activity Diagram

Alur Pertama dalam activity diagram Penerimaan Peserta Didik Baru (PPDB) adalah *Admin* membuka pendaftaran lalu sistem akan membuka pendaftaran sehingga para calon siswa dapat mulai mendaftar pada aplikasi PPDB, setelah mendaftar calon siswa melengkapi identitas diri, *upload* berkas-berkas dan memilih jalur pendaftaran yang ingin diambil, lalu admin menseleksi seluruh siswa yang telah mendaftar, lalu sistem akan menyimpan hasil seleksi tersebut, jika tidak lolos seleksi calon siswa berarti tidak diterima menjadi siswa baru, jika diterima calon siswa harus melakukan daftar ulang, kemudian admin akan memverifikasi berkas-berkas yang dibawa oleh calon siswa. Jika tidak lolos verifikasi berarti calon siswa tidak diterima menjadi siswa baru, bila lolos seleksi berarti siswa diterima menjadi siswa baru.



Gambar 2. Activity Diagram Aplikasi PPDB

Desain Web Service

Desain sistem *web service* REST API bertujuan untuk merancang mengenai fungsi-fungsi yang akan menjadi penerapan basis API dengan menerapkan metode REST. Desain API ini berdasarkan analisis kebutuhan yang nantinya API tersebut akan di implementasikan pada aplikasi berbasis *mobile*. API yang akan dirancang hanya sebatas API untuk *user* atau calon siswa saja hal ini mencakup pada saat calon siswa mulai proses pendaftaran hingga pada saat pengumuman penerimaan dan daftar ulang. Berikut ini adalah desain REST API untuk

aplikasi PPDB *online*, dapat dilihat pada Tabel 1.

Tabel 1
Desain *Web Service* REST API

No	Proses	Rancangan API	
		Method	URI
1	Login user	POST	/api/auth/login/
2	Registrasi user	POST	/api/auth/register/
3	Logout	POST	/api/auth/logout/
3	Menampilkan profile siswa	GET	/api/profile/
4	Mengedit / melengkapi api identitas siswa	PUT / PATCH	/api/profile/
5	Upload berkas-berkas siswa	PUT	/api/profile/berkas/
6	Melihat Jadwal Kegiatan PPDB	GET	/api/list_events/
7	Melihat Notifikasi Siswa	GET	/api/notifikasi/

Implementasi *Web Service*

a) LoginAPI

Login API dapat diakses melalui URL `/api/auth/login` menggunakan *method POST* dengan memberikan data *email* dan *password* pada *body*. Untuk hasil *response* data dari *login* API dapat dilihat pada Gambar 3.

```

1 {
2   "user": {
3     "email": "user5@gmail.com",
4     "first_name": "John",
5     "last_name": "Afrizal"
6   },
7   "token": "2fd32df94806519cee308ee9a14da4e4c7e19ef5814045e8c5df688175352c2"
8 }

```

Gambar 3. Response data LoginAPI

b) RegisterAPI

Register API dapat diakses melalui URL `/api/auth/register` menggunakan *method POST* dengan memberikan data *first_name*, *last_name*, *email* dan *password* pada *body*. Untuk hasil *response* data dari *register* API dapat dilihat pada Gambar 4.

```

2   "user": {
3     "email": "ryan@gmail.com",
4     "first_name": "Ryan",
5     "last_name": "Afrizal"
6   },
7   "nis": "2021015",
8   "jenis_kelamin": null,
9   "tanggal_lahir": null,
10  "tempat_lahir": null,
11  "umur": null,
12  "alamat": null,
13  "asal_sekolah": null,
14  "status": 0,
15  "nilai_matematika": null,
16  "nilai_indonesia": null,
17  "nilai_inggris": null,
18  "nilai_ipa": null,
19  "foto_diri": null,
20  "berkas_ijazah": null,
21  "berkas_akta": null,
22  "berkas_kesehatan": null,
23  "berkas_tambahan": null

```

Gambar 4. Response data RegisterAPI

c) ProfileSiswaAPI

Profile API dapat diakses melalui URL `/api/profile` menggunakan *method PUT*, atau *PATCH* dengan memberikan *token authentication* pada *header authentication*. Untuk hasil *response* data dari *profile* API dapat dilihat pada Gambar 5.

```

2   "user": {
3     "email": "ryan@gmail.com",
4     "first_name": "Ryan",
5     "last_name": "Afrizal"
6   },
7   "nis": "2021015",
8   "jenis_kelamin": "L",
9   "tanggal_lahir": "2006-04-20",
10  "tempat_lahir": "Bandung",
11  "umur": 18,
12  "alamat": "Perumahan Cikawao, Nagrak, Pacet, Kabupaten Bandung",
13  "asal_sekolah": null,
14  "status": 1,
15  "nilai_matematika": null,
16  "nilai_indonesia": null,
17  "nilai_inggris": null,
18  "nilai_ipa": null,
19  "foto_diri": "/media/images/2021015/foto_diri.jpg",
20  "berkas_ijazah": null,
21  "berkas_akta": null,
22  "berkas_kesehatan": null,
23  "berkas_tambahan": null

```

Gambar 5. Response data ProfileSiswaAPI

d) UploadBerkasAPI

UploadBerkasAPI dapat diakses melalui URL `/api/profile/berkas` menggunakan *method PUT* dengan memberikan *token authentication* pada *header authentication*. Untuk hasil *response* data dari *UploadBerkasAPI* dapat dilihat pada Gambar 6.

```

2   "user": {
3     "email": "ryan@gmail.com",
4     "first_name": "Ryan",
5     "last_name": "Afrizal"
6   },
7   "nis": "2021015",
8   "jenis_kelamin": "L",
9   "tanggal_lahir": "2006-04-20",
10  "tempat_lahir": "Bandung",
11  "umur": 18,
12  "alamat": "Perumahan Cikawao, Nagrak, Pacet, Kabupaten Bandung",
13  "asal_sekolah": null,
14  "status": 2,
15  "nilai_matematika": "40.00",
16  "nilai_indonesia": "20.00",
17  "nilai_inggris": "10.00",
18  "nilai_ipa": "33.00",
19  "foto_diri": "/media/images/2021015/foto_diri.jpg",
20  "berkas_ijazah": "/media/images/2021015/file_ijazah.jpg",
21  "berkas_akta": "/media/images/2021015/file_akta.jpg",
22  "berkas_kesehatan": "/media/images/2021015/file_kesehatan.jpg",
23  "berkas_tambahan": null

```

Gambar 6. Response data UploadBerkasAPI

e) PengajuanAPI

PengajuanAPI dapat diakses melalui URL `/api/profile/pengajuan` menggunakan *method PUT* dengan memberikan *token authentication* pada *header authentication*. Untuk hasil *response* data dari *PengajuanAPI* dapat dilihat pada Gambar 7.

```

Body ▾ 200 OK 78 ms
Pretty Raw Preview Visualize JSON
2   "user": {
3     "email": "ryan@gmail.com",
4     "first_name": "Ryan",
5     "last_name": "Afrizal"
6   },
7   "nis": "2021015",
8   "jenis_kelamin": "L",
9   "tanggal_lahir": "2006-04-20",
10  "tempat_lahir": "Bandung",
11  "umur": 18,
12  "alamat": "Perumahan Cikawao, Nagrak, Pacet, Kabupaten Bandung",
13  "asal_sekolah": null,
14  "status": 11,
15  "nilai_matematika": "40.00",
16  "nilai_indonesia": "20.00",
17  "nilai_inggris": "10.00",
18  "nilai_ipa": "33.00",
19  "foto_diri": "/media/images/2021015/foto_diri.jpg",
20  "berkas_ijazah": "/media/images/2021015/file_ijazah.jpg",
21  "berkas_akta": "/media/images/2021015/file_akta.jpg",
22  "berkas_kesehatan": "/media/images/2021015/file_kesehatan.jpg",
23  "berkas_tambahan": "/media/images/2021015/file_afirmasi.jpg"

```

Gambar 7. Response data PengajuanAPI

f) NotifikasiAPI

NotifikasiAPI dapat diakses melalui URL `/api/notifikasi` menggunakan `method GET` dengan memberikan `token authentication` pada `header authentication`. Untuk hasil `response data` dari NotifikasiAPI dapat dilihat pada Gambar 8.

```

Body ▾ 200 OK
Pretty Raw Preview Visualize JSON
1   [
2     {
3       "id": 6,
4       "notifikasi": "Verifikasi Berkas Berhasil",
5       "tanggal_notifikasi": "2020-07-21"
6     }
7   ]

```

Gambar 8. Response data NotifikasiAPI

g) ListKegiatanAPI

ListKegiatan dapat diakses melalui URL `/api/list_events` menggunakan `method GET` dengan memberikan `token authentication` pada `header authentication`. Untuk hasil `response data` dari ListKegiatanAPI dapat dilihat pada Gambar 9.

```

Body ▾ 200 OK
Pretty Raw Preview Visualize JSON
1   [
2     {
3       "id": 5,
4       "name": "Pembukaan Pendaftaran",
5       "start_date": "2020-07-14",
6       "end_date": "2020-07-21"
7     },
8     {
9       "id": 6,
10      "name": "Proses Seleksi Siswa",
11      "start_date": "2020-07-22",
12      "end_date": "2020-07-24"
13    },
14    {
15      "id": 7,
16      "name": "Pengumuman Penerimaan",
17      "start_date": "2020-07-25",
18      "end_date": "2020-07-25"
19    }
20  ]

```

Gambar 9. Response data ListKegiatanAPI

4. Penutup Kesimpulan

Berikut ini adalah beberapa kesimpulan yang diambil dari penelitian yang telah dilakukan :

1. Dengan adanya sistem aplikasi pendaftaran peserta didik baru (PPDB) berbasis *online*, dapat mempermudah calon siswa atau wali siswa untuk mendaftar atau mendaftarkan anaknya ke sekolah yang dituju.
2. Dengan mengimplementasikan *web service* berbasis REST pada sistem aplikasi berbasis *mobile* dan *web* dapat membuat kedua aplikasi tersebut terhubung satu sama lain.

Saran

Berdasarkan kesimpulan yang penulis kemukakan di atas, maka penulis memberikan beberapa saran yang dapat membantu agar penelitian ini dapat berkembang menjadi lebih baik:

1. Membuat aplikasi berbasis *mobile* untuk *admin* atau pihak sekolah, agar mempermudah *admin* atau pihak sekolah dalam penyeleksian calon siswa.
2. Membuat API (*Application Programming Interface*) untuk *admin* agar dapat mempermudah dalam pembuatan aplikasi berbasis *mobile* untuk *admin* atau pihak sekolah

Referensi

- Angga Diputra, I. P. W., Raditya Putra, I. G. L. A., & Satwika, I. P. (2018). SISTEM INFORMASI PKK BERBASIS WEBSITE DENGAN FITUR MOBILE DI KELURAHAN PENATIH KOTA DENPASAR. ISSN: 2614-8722. *Jurnal Teknologi Informasi MURA*, 10(2), 92. Diambil dari: <http://jurnal-stmik.muralinggau.ac.id/index.php/jti/article/view/389> (16 April 2020)
- Hadinata, N. S., & Fitriyani. (2017). Website Pengolahan Nilai Siswa Menggunakan Konsep Model-View-Controller pada LIA Pamulang. ISSN: 2477-3786. *Informatika*, 4(2), 2–7. Diambil dari : <http://ejournal.uigm.ac.id/index.php/IG/article/view/746> (12 Mei 2020)
- Kapalomen. (2016). Django : Pola MVC Dan MVT Pada Framework Django. Diambil dari : <https://www.kapalomen.com/2016/11/django-pola-mvc-dan-mvt-pada-framework-django.html> (14 Mei 2020)
- Kiki. (2020). 68 Link Resmi Pendaftaran SIAP PPDB Online 2020/2021. Diambil dari : <https://mamikos.com/info/68-link-resmi-pendaftaran-siap-ppdb-online-2020-2021/> (10 Mei 2020)
- Muhardian, A. (2016). *Belajar Django #1: Pengenalan Dasar Django untuk Pemula*. Diambil dari : <https://www.petanikode.com/django-untuk-pemula/> (20 Maret 2020)
- NATALIANA, D., SYAMSU, I., & GIANTARA, G. (2014). Sistem Monitoring Parkir Mobil menggunakan Sensor Infrared berbasis RASPBERRY PI. ISSN: 2338-8323. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 2(1), 68. Diambil dari: <https://ejurnal.itenas.ac.id/index.php/elkomika/article/view/819/1018> (06 Juni 2020)
- Rulloh, A., Mahmudah, D. E., & Kabetta, H. (2017). Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android. ISSN: 2598-2958. *Teknikom: Teknologi Informasi, Ilmu Komputer dan Manajemen*, 1(2), 85–89. Diambil dari: <http://journal.swu.ac.id/index.php/teknikom/article/view/50> (05 Mei 2020)
- Sari, A. U. W., Imron, A., & Sobri, A. Y. (2016). *Efektivitas Penerimaan Peserta Didik Baru (PPDB) Melalui Sistem Penerimaan Peserta Didik Online*. ISSN: 2716-0122. 1–11. Diambil dari : <http://repository.unhas.ac.id/handle/123456789/17968> (07 April 2020)
- Sibagariang, S. (2016). Web Service Rest Pada Perpustakaan Berbasis Android. *Maharjana Informasi*, 1(2), 8–11. ISSN: 2527-8290 <http://e-journal.sari-mutiara.ac.id/index.php/7/article/view/172>. Diambil dari : <http://e-journal.sari-mutiara.ac.id/index.php/7/article/view/91> (15 Mei 2020)
- Susanto, R., & Andriana, A. D. (2016). Perbandingan Model Waterfall dan Prototyping Untuk Pengembangan Sistem Informasi. ISSN: 1411-9374. *Teknikom: Teknologi Informasi, Ilmu Komputer dan Manajemen*, 41–46. Diambil dari : <https://repository.unikom.ac.id/30459/> (12 Mei 2020)
- Sy, H., & Intan, I. (2019). *Implementasi Restful Api Portal Akademik Stmik Dipanegara Berbasis Android*. ISSN: 2354-6999 *VIII(2)*, 63–70. Diambil dari : <https://ejurnal.dipanegara.ac.id/index.php/sisiti/article/view/27-40> (04 April 2020)