

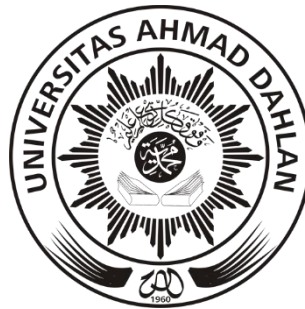
Sistem Informasi untuk Verifikasi dan Validasi Data DAPODIK Menggunakan Metode RAD (*Rapid Application Development*)

**Rui Faqri
1800016004**

Skripsi diajukan kepada
Fakultas Sains dan Teknologi Terapan
sebagai bagian persyaratan untuk meraih derajat

Sarjana Komputer

pada Program Studi Sistem Informasi



Universitas Ahmad Dahlan

Yogyakarta

Oktober, 2022

Daftar Isi

| | |
|--------------------------------------|----------|
| Daftar Isi..... | ii |
| Daftar Gambar..... | v |
| Daftar Tabel | vi |
| Bab 1..... | 1 |
| Pendahuluan | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Identifikasi Masalah | 4 |
| 1.3 Ruang Lingkup | 5 |
| 1.4 Rumusan Masalah | 6 |
| 1.5 Tujuan Penelitian..... | 6 |
| 1.6 Manfaat Penelitian..... | 6 |
| Bab 2..... | 7 |
| Kajian Teori..... | 7 |
| 2.1 Kajian Pustaka Terdahulu..... | 7 |

| | | |
|-------------------|---|-----------|
| 2.2 | Dasar Teori | 13 |
| 2.2.1 | Sistem Informasi | 13 |
| 2.2.2 | Basis Data | 16 |
| 2.2.3 | API DAPODIK | 19 |
| 2.2.4 | Framework Angular | 21 |
| 2.2.5 | Postman..... | 23 |
| 2.2.6 | Flow Map | 26 |
| 2.2.7 | Data Flow Diagram (DFD) | 27 |
| 2.2.8 | Entity Relationship Diagram (ERD)..... | 29 |
| 2.2.9 | Flowchart | 30 |
| 2.2.10 | Metode Rapid Aplication Development (RAD) | 31 |
| Bab 3 | | 33 |
| Metodologi | | 33 |
| 3.1 | Subjek Penelitian | 33 |
| 3.2 | Alat Penelitian | 34 |
| 3.2.1 | Perangkat Keras (Hardware) | 34 |
| 3.2.2 | Perangkat Lunak (Software) | 34 |
| 3.3 | Metode Pengembangan Sistem | 35 |

| | |
|--|-----------|
| 3.3.1 Fase Perencanaan Syarat-Syarat | 35 |
| 3.3.2 Perancangan Sistem (Workshop Design) | 36 |
| 3.3.3 Implementasi Sistem (Coding & Testing) | 36 |
| 3.4 Tahap Penelitian | 37 |
| 3.5 Jadwal Penelitian | 38 |
| Daftar Acuan | 40 |

Daftar Gambar

| | |
|--|----|
| Gambar 2.1 Sistem informasi sederhana..... | 14 |
| Gambar 2.2 Sistem informasi..... | 15 |
| Gambar 2.3 Angular First Run..... | 22 |
| Gambar 2.4 Dashboard Postman..... | 24 |
| Gambar 2.5 Metode RAD (Kendall, 2003)..... | 31 |
| Gambar 3.1 Tahap Penelitian..... | 37 |

Daftar Tabel

| | |
|--|----|
| Tabel 2.1 Kajian Peneliti Terdahulu | 10 |
| Tabel 2.2 Simbol Class Diagram | 26 |
| Tabel 2.3 Simbol Data Flow Diagram | 28 |
| Tabel 2.4 Notasi Entity Relationship Diagram | 29 |
| Tabel 2.5 Simbol Flowchart..... | 30 |
| Tabel 3.1 Spesifikasi Laptop..... | 34 |
| Tabel 3.2 Spesifikasi Software Penelitian..... | 34 |
| Tabel 3.3 Jadwal Penelitian..... | 38 |

Bab 1

Pendahuluan

1.1 Latar Belakang

Keberadaan teknologi kini telah menjadi kebutuhan yang menunjang bagi kehidupan manusia di era modern ini. Hal tersebut dapat dilihat dengan dibutuhkannya sebuah sistem untuk mendukung aktifitas manusia sehari-hari. Pertukaran data yang cepat, efisien dan akurat merupakan keunggulan teknologi modern. Metode ini dapat diaplikasikan untuk mendapatkan informasi atau sumber daya dari berbagai sumber data. Salah satu bentuk penerapan teknologi berada pada sistem informasi, diantara teknologi sistem informasi yaitu website, aplikasi mobile dan lain-lain (Kurniawan, 2019). Dalam setiap sistem informasi tersebut memiliki struktur yang berbeda-beda diantaranya bahasa pemrograman atau framework (Ramirez, 2019) yang digunakan, database (Khan, 2019), UI/UX (Kyi, 2016), dan lain-lain (Utomo, 2020). Sistem informasi mampu membuat berbagai pekerjaan menjadi lebih efektif dan efisien (Komalasari, 2018).

Data Pokok Pendidikan (DAPODIK) adalah sistem pendataan skala nasional yang terpadu, dan merupakan sumber data utama pendidikan nasional, yang merupakan bagian dari Program perencanaan pendidikan nasional dalam mewujudkan insan Indonesia yang cerdas dan kompetitif. Dengan demikian sudah sangat penting dalam sistem pendidikan harus ada yang namanya DAPODIK ini. Dengan adanya DAPODIK ini perencanaan pendidikan akan menjadi matang dan berkualitas. Karena jika perencanaan pendidikan tidak matang, maka seluruh program yang telah terbentuk dari perencanaan tersebut akan jauh dari apa yang diharapkan. Demi melaksanakan perencanaan dan juga melakukan program-program pendidikan agar tepat pada sasaran, maka dibutuhkan data yang cepat, lengkap, valid, akuntabel dan terus up to date. Adanya data yang cepat, lengkap, valid, akuntabel dan up to date tersebut, maka seluruh proses perencanaan, pelaksanaan, pelaporan dan evaluasi kinerja program-program pendidikan nasional dapat dilaksanakan dengan lebih terukur, tepat sasaran, efektif, efisien dan berkelanjutan. Sehubungan dengan hal tersebut, maka Departemen Pendidikan Nasional telah mengembangkan suatu sistem pendataan skala nasional yang terpadu dan disebut dengan data pokok pendidikan atau yang lebih dikenal dengan DAPODIK.

Meskipun secara *up to date* data pendidikan oleh operator DAPODIK di jalankan dengan baik namun belum semua data yang dimaksudkan (*input*) ke aplikasi DAPODIK akan diterima oleh server dapodik. Sehingga sampai dengan saat ini perlu kita ketahui bersama bahwa DAPODIK di gunakan secara Nasional untuk meng *up date* seluruh

data pendidikan di seluruh Indonesia, kenyataan saat ini aplikasi DAPODIK hanya menghasilkan data yang sesuai dikirim oleh operator sekolah sehingga kenyataannya guru masih banyak yang mengeluh dengan data mereka yang selalu berpengaruh pada penyaluran berbagai macam aneka tunjangan yang di berikan oleh pemerintah, hal ini dikarenakan guru yang tidak dapat menggunakan aplikasi DAPODIK. Hal tersebut bukan tanpa alasan, dimana para guru tersebut tidak paham cara menggunakannya karena terlalu kompleks, tidak paham tentang IT, dan akses jaringan yang sulit. Maka diperlukan sebuah aplikasi sederhana yang dapat mengambil data tertentu dari dalam aplikasi DAPODIK, hal ini bertujuan agar guru dapat melakukan pengecekan data dengan mudah. Kondisi dan permasalahan diatas menunjukkan bahwa DAPODIK belum maksimal baik dari sisi input data, output data maupun verifikasi data. Hal ini perlu dikaji dan dianalisis untuk mencari solusi pemanfaatan DAPODIK dalam menunjang administrasi sekolah.

Dalam pengembangan perangkat lunak terdapat beberapa tahapan tertentu yang dikenal sebagai metodologi pengembangan perangkat lunak (Parahita, 2018). Contoh metodologinya yaitu Agile (Jain, 2018), Water Fall (Purnia, 2019), Lean Development (Manuaba, 2019), TDD (*Test- Driven Development*) (Hellmann, 2010), BDD (*Behavior-Driven Development*) (Abushama, 2020), T-BDD (*Test Behavior- Driven Development*) dan banyak lagi yang telah diterapkan sebagai pedoman untuk proses pengembangan perangkat lunak (Manuaba, 2019).

Pada penelitian ini menggunakan metode RAD sebagai bagian dari ADMMC (*Agile based development methodology for mobile commerce applications*) lebih bersifat aplikatif, praktis dan mudah menyesuaikan untuk kebutuhan sistem perangkat bergerak dibandingkan dengan model agile lainnya (Kosasi, 2015). Pemilihan metode ini bukan tanpa alasan, dimulai dari penerapannya yang mudah seperti memulai proses pengembangan perangkat lunak sesuai dengan tahapan. Setiap tahapan dianggap selesai setelah syarat terpenuhi, tetapi jika belum memenuhi maka harus mengulang tahap tersebut . Hal ini untuk meminimalisir terdapatnya kesalahan atau bug yang seharusnya sudah teratasi pada sebuah tahap pengembangan perangkat lunak. Berdasarkan penjelasan dan permasalahan yang ditemukan, maka penelitian ini mengajukan judul **“Sistem Informasi untuk Verifikasi dan Validasi Data DAPODIK Menggunakan Metode RAD (Rapid Application Development)”**.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dijabarkan di atas, maka permasalahan yang dapat diangkat dalam penelitian ini adalah sebagai berikut:

1. Terdapat data siswa, guru, ataupun sekolah yang tidak sesuai pada aplikasi DAPODIK.
2. Guru tidak dapat sepenuhnya mengakses dan memperbaiki data individu di DAPODIK.

3. Dibutuhkan sebuah aplikasi untuk mengambil dan melakukan perbaikan data DAPODIK yang tidak sesuai.
4. Aplikasi tersebut dapat digunakan oleh guru dan operator untuk melakukan pengecekan data, tanpa harus mengakses aplikasi DAPODIK secara langsung.

1.3 Ruang Lingkup

Berdasarkan penjelasan pada latar belakang sebelumnya, dan memfokuskan pembahasan pada masalah yang dibahas, maka dibuat ruang lingkup yaitu:

1. Aplikasi DAPODIK merupakan system pendataan yang digunakan untuk administrasi sekolah.
2. Membuat system sederhana untuk mengambil data dari aplikasi DAPODIK.
3. System tersebut dapat digunakan oleh guru untuk memperbaiki data yang tidak sesuai di DAPODIK.
4. System tersebut dapat digunakan oleh operator, dimana operator dapat melakukan pengecekan data yang tidak sesuai dan melakukan update ke aplikasi DAPODIK.
5. System akan memanfaatkan API DAPODIK.
6. System akan dikembangkan menggunakan framework Angular.
7. Pengujian pengambilan data akan menggunakan Postman.

1.4 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan diatas, maka rumusan masalah yang dapat diajukan dalam penelitian ini adalah bagaimana merancang dan membangun sistem informasi untuk memverifikasi dan memvalidasi data DAPODIK dengan metode RAD?

1.5 Tujuan Penelitian

Tujuan yang hendak dicapai pada penelitian ini adalah untuk merancang dan membangun sistem informasi untuk verifikasi dan validasi data DAPODIK dengan metode RAD.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1. Pengembangan Sistem Informasi untuk verifikasi dan validasi data DAPODIK dengan memanfaatkan API DAPODIK dapat mempermudah dalam melakukan pengambilan dan pengecekan data.
2. Hasil penelitian ini diharapkan dapat menjadi referensi yang bermanfaat bagi dunia akademik khususnya dalam penelitian yang dilakukan oleh peneliti tentang pengembangan sistem informasi DAPODIK.

Bab 2

Kajian Teori

2.1 Kajian Pustaka Terdahulu

Pertama, penelitian sebelumnya oleh (Rulloh, 2017), Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android. Penelitian ini memanfaatkan layanan web RESTful dimana aplikasi saat ini terhubung ke jaringan global. Dengan dukungan format notasi dari JSON, hasil dari data akan lebih cepat dimuat di aplikasi dan memungkinkan pengembang untuk mengenali objek yang dihasilkan. Pertama, membuat database untuk menyimpan konten yang mendukung aplikasi. Kemudian buat pengontrol yang digunakan untuk setiap panggilan API aktif setiap konten yang diperlukan. Setelah itu aplikasi di android dibuat dengan dukungan beberapa library, yaitu: Volley untuk memanggil API dan HTML-TextView untuk mengonversi data konten dalam format HTML ke dalam format TextView pada perangkat android. Penelitian ini telah menghasilkan bahwa REST dapat menjadi diterapkan dalam Aplikasi Bimbingan ini.

Kedua, penelitian sebelumnya oleh (Pranata, 2018), Perancangan *Application Programming Interface* (API) Berbasis Web Menggunakan Gaya Arsitektur

Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit. Penelitian ini mengimplementasikan REST dalam mengembangkan API sebagai back-end perawatan kulit sistem informasi pasien klinik. API dikembangkan menggunakan Javascript Object Notation (JSON) sebagai format standar untuk komunikasi data dan JSON Web Token (JWT) sebagai otentikasi pengguna kode. API diuji dalam tiga tahap: pengujian JWT pada beberapa server back end, pengujian API dengan Equivalence Partitioning dan pengujian fungsional sistem. Dalam penelitian ini gaya arsitektur REST berhasil diimplementasikan pada sistem administrasi pasien klinik perawatan kulit. Kemudian pengujian otentikasi REST API dalam penelitian ini memperoleh keberhasilan penuh karena JWT telah dimengerti oleh semua *back-end server*. Begitu pula untuk pengujian masukan sistem dengan metode equivalence partitioning dan pengujian fungsional sistem.

Ketiga, penelitian sebelumnya oleh (Putri, 2018), Implementasi Metode Rapid Application Development Pada *Website Service Guide “Waterfall Tour South Sumatera”*. Potensi objek-objek wisata air terjun di seluruh provinsi Indonesia belum digali secara optimal, begitu juga dengan promosi belum banyak dilakukan apalagi dengan menggunakan media online atau digital dikarenakan belum adanya website khusus promosi yang lengkap tentang objek wisata air terjun yang ada di Provinsi Sumatera Selatan. Maka perlu melakukan implementasi metode rapid application development Pada *Website Service Guide “Waterfall Tour South Sumatera”*. Metode

pembuatan aplikasi ini menggunakan metode perancangan sistem *Rapid Application Development* (RAD), yang dimulai dengan tahapan requirement planning, system design, dan implementation. Dengan menggunakan tahapan metode RAD dalam pembangunan website dapat menghasilkan sebuah website yang memberikan informasi yang objektif, sehingga dapat membantu masyarakat dalam mencari informasi wisata air terjun di Sumatera Selatan.

Keempat, penelitian sebelumnya oleh (Sembara, 2020), Rancang Bangun *Application Programming Interface* (API) Menggunakan Gaya Arsitektur Representational State Transfer (REST) Untuk Pengembangan Sistem Informasi Chatting. PT. Vascomm Solusi Teknologi adalah sebuah perusahaan yang bergerak pada bidang IT solutions dan start up factory. Pada penelitian ini bertujuan melakukan riset tentang teknologi API dengan gaya arsitektur REST dan penggunaan socket.io pada pengembangan sistem informasi chatting. Berdasarkan hasil penelitian dapat disimpulkan bahwa, perancangan sistem menggunakan API dengan gaya arsitektur REST dapat diimplementasikan pada perancangan sistem berbasis API dengan gaya arsitektur REST. Hal ini bertujuan untuk menjadikan sebuah sistem memiliki performa yang baik, cepat dan mudah untuk dikembangkan terutama dalam pertukaran dan komunikasi data.

Kelima, penelitian sebelumnya oleh (Julio, 2021), Implementasi API Payment Gateway Menggunakan Arsitektur Microservice. Penelitian ini bertujuan untuk melakukan integrasi pembayaran online, PT. Emporia Digital Raya perlu membuat

aplikasi *Payment Gateway* yang dapat meneruskan pembayaran ke bank tujuan. Arsitektur microservice dapat diterapkan untuk mengatasi masalah yang ada pada arsitektur monolithic dimana pengembangan aplikasi dilakukan dalam bentuk web service kecil yang saling berkomunikasi satu dengan yang lain. Dengan menggunakan konsep routing, *payment gateway* memiliki tanggung jawab seperti router dimana *payment gateway* berperan mencari rute ke bank mana data akan dikirimkan. *Payment gateway* akan memproses request dan membaca *Fields Beneficiary Bank Code* untuk menacari rute. Jika bank tujuan tidak ditemukan maka *payment gateway* akan menjalankan query dan mencari rute berdasarkan prioritas yang telah ditentukan oleh PT. Emporia Digital Raya.

Tabel 2.1 Kajian Peneliti Terdahulu

| Kriteria | Kajian 1 | Kajian 2 | Kajian 3 | Kajian 4 | Kajian 5 |
|-----------------|---|--|--|--|--|
| Peneliti | Amin Rulloh, Dewi Erla Mahmudah, Herman Kabetta | Beni Adi Pranata, Astria Hijriani, Akmal Junaidi | Meidyan Permata Putri, Hendra Effendi | Sebastianus Sembara | Erry Julio, Magdalena A. Ineke Pakereng |
| Tahun | 2017 | 2018 | 2018 | 2020 | 2021 |
| Judul | Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android | Perancangan Application Programming Interface (API) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit | Implementasi Metode Rapid Application Development Pada Website Service Guide "Waterfall Tour South Sumatera" | Rancang Bangun Application Programming Interface (API) Menggunakan Gaya Arsitektur Representational State Transfer (REST) Untuk Pengembangan Sistem Informasi Chatting | Implementasi API Payment Gateway Menggunakan Arsitektur Microservice |
| Objek | Aplikasi Panduan Kepaskibraan | Sistem Informasi Administrasi | Website | PT. Vascomm Solusi Teknologi | PT. Emporia Digital Raya |

| | | | | | |
|-------------------------------------|---|---|--|--|---|
| | Berbasis Android | Pasien Klinik Perawatan Kulit | Service Guide "Waterfall Tour South Sumatera" | | |
| Metode/ Teori yang digunakan | REST | Waterfall | RAD | REST | Agile |
| Hasil | Dengan digunakannya REST API dalam pembuatan Aplikasi Panduan Kepaskibraan dan dengan adanya dukungan notasi JSON maka memudahkan dalam proses penerapan teknologi pada aplikasi yang dibangun. | Pengujian otentikasi REST API, metode equivalence partitioning dan pengujian fungsional sistem. keberhasilan penuh karena JWT telah dimengerti oleh semua back-end server | Sistem dapat menjadi media promosi dan memperkenalkan wisata air terjun kepada masyarakat luas. Hasil pengujian menampilkan sebuah website yang dapat memberikan informasi yang objektif sehingga dapat membantu dan memberikan kemudahan bagi pengunjung dalam mencari informasi wisata Air terjun di Sumatera Selatan. | Pengembangan sistem berbasis API dapat mempercepat pembuatan sebuah sistem | Penelitian ini membangun aplikasi payment gateway menggunakan arsitektur microservice, dimana peneliti dapat memelihara aplikasi berdasarkan pembagian bank. Hal ini dapat meminimalisir kesalahan dalam melakukan pengembangan maupun pemeliharaan aplikasi. Selain itu, jika payment gateway ingin melakukan integrasi dengan bank baru maka, peneliti hanya perlu membuat microservice baru. |

Dari **Tabel 2.1** diatas dapat disimpulkan bahwa pengembangan perangkat lunak tidak dapat dilakukan sembarangan atau asal jadi saja. Kita perlu memperhatikan beberapa aspek agar pengembangan perangkat lunak dapat berjalan dengan baik dan sesuai dengan kebutuhan kita. Dalam pengembangan perangkat lunak terdapat metodologi yang dapat

digunakan seperti Agile, Water Fall, Lean Development, TDD (Test- Driven Development), BDD (*Behavior-Driven Development*), T-BDD (*Test Behavior- Driven Development*) dan banyak lagi yang telah diterapkan sebagai pedoman untuk proses pengembangan perangkat lunak. Untuk penelitian ini kita akan membahas metode RAD (*Rapid Aplication Development*), dalam beberapa penelitian dan studi kasus yang berbeda. Pada penelitian pertama yang berobjek pada *Website Service Guide “Waterfall Tour South Sumatera”*, dengan menggunakan metode RAD pengembangan sistem dilakukan dan bertujuan untuk mempromosikan wisata air terjun yang ada di Sumatra Selatan yang berbasis digital. Pada penelitian kedua objek yang digunakan adalah Aplikasi Panduan Kepaskibraan Berbasis Android, dengan menggunakan metode REST API dapat membantu mempermudah pengaplikasian teknologi kepada aplikasi yang sedang dibangun. Pada penelitian ketiga yang berobjek Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit, pengujian otentikasi REST API dalam penelitian ini memperoleh keberhasilan penuh karena JWT telah dimengerti oleh semua *back-end server*. Begitu pula untuk pengujian masukan sistem dengan metode equivalence partitioning dan pengujian fungsional sistem. Pada penelitian keempat yang berobjek PT. Vascomm Solusi Teknologi, dimana pemanfaatan API dapat membantu dalam mempercepat pembangunan sebuah sistem atau aplikasi. Pada penelitian kelima yang berobjek PT. Emporia Digital Raya, dimana aplikasi *payment gateway* yang dibangun menggunakan arsitektur microservice dapat melakukan pengembangan atau pemeliharaan berdasarkan pembagian bank dan jika terdapat bank baru yang ingin terhubung maka hanya perlu membangun microservice baru.

Berdasarkan penelitian terdahulu sudah banyak tim pengembang yang menggunakan API sebagai topik penelitian dan menggunakan berbagai macam metode seperti RAD, Waterfall, REST. Agile, dll. Metode pengembangan perangkat lunak RAD (Rapid Application Development) memberikan dampak yang bagus kepada para pengembang. Tujuan yang hendak dicapai dari penelitian ini adalah perancangan sistem informasi sinkronisasi data DAPODIK menggunakan metode RAD dengan memanfaatkan API DAPODIK.

2.2 Dasar Teori

2.2.1 Sistem Informasi

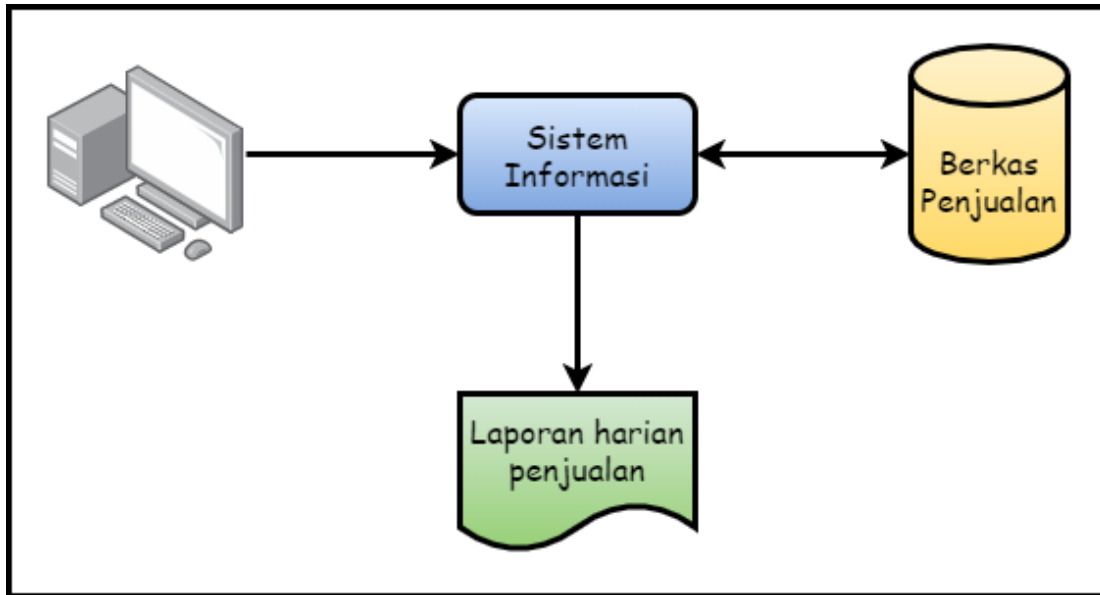
Pada saat ini setiap kegiatan manusia selalu berkaitan dengan sistem informasi. Sistem informasi diterapkan pada berbagai bidang seperti kantor, pasar swalayan, bandara, ataupun di rumah saat kita menggunakan sebuah aplikasi pada sebuah *smartphone*.

Traveloka menyediakan reservasi pesawat terbang yang digunakan untuk melayani pembelian tiket yang bersifat *online*. Dengan menggunakan aplikasi web, orang yang sedang melakukan perjalanan dengan menggunakan pesawat terbang dapat melakukan pembelian tiket secara langsung.

Sebuah sistem informasi tidak harus selalu kompleks. Sebuah sistem tidak harus selalu rumit ada juga sistem yang sederhana. Sebagai contoh dimana sistem digunakan untuk

mencatat transaksi penjualan dan hanya melibatkan satu orang saja Seperti pada

Gambar 2.1.



Gambar 2.1 Sistem informasi sederhana

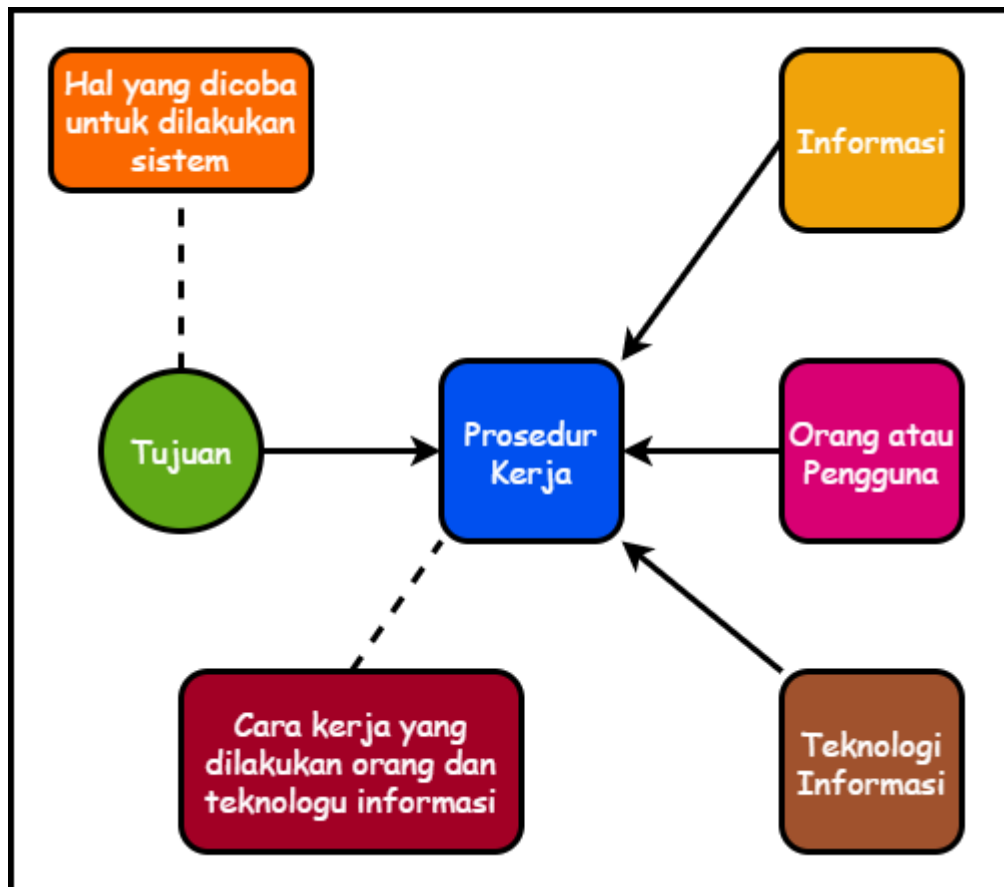
Sebuah sistem informasi yang kompleks, terdapat banyak pengguna dan jaringan yang tersebar luas, hal ini bertujuan agar pengguna dapat menggunakan sebuah sistem informasi dari berbagai tempat. Kemampuan yang dapat dilakukan oleh sistem informasi diantaranya:

1. Melaksanakan komputasi numerik, bervolume besar, dan berkecepatan tinggi.
2. Mampu menjadi media komunikasi.
3. Dapat menyimpan informasi yang besar dan mudah diakses.
4. Menjadi media informasi yang dapat diakses dari seluruh dunia dengan cepat dan mudah.
5. Meningkatkan efektivitas dan efisiensi dalam pekerjaan.

6. Menyajikan informasi dengan jelas.
7. Membuat sebuah pekerjaan yang dilakukan secara manual menjadi otomatis.
8. Mempercepat dalam hal pengetikan dan penyuntingan.

Menurut Alter (1992) “Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi”. Dari penjelasan Alter tersebut diilustrasikan seperti

Gambar 2.2.



Gambar 2.2 Sistem informasi

Sistem informasi tidak selalu berhubungan dengan komputer. Pekerjaan yang menggunakan sistem informasi komputer disebut Sistem Informasi Berbasis Komputer (Computer Based Information Systems atau CBIS). Sistem informasi mencakup beberapa komponen utama yaitu manusia, komputer, prosedur kerja, teknologi informasi, proses, dan bertujuan untuk mencapai tujuan atau *goals*.

2.2.2 Basis Data

Sistem informasi memiliki sebuah media untuk menyimpan dan mengelola sebuah data yang disebut basis data. Basis data dapat mempercepat upaya layanan pelanggan menyediakan informasi yang cepat dan akurat untuk memantu memuat keputusan pemecahan masalah segera erdasarkan informasi yang tersedia. Banyak aplikasi berbasis database termasuk: semua perbankan, pemesanan dan perencanaan penerbangan penerimaan universitas dan proses entri data, pemrosesan pesanan penjualan, pembelian dan pendaftaran data barang di perusahaan komersial, pendaftaran data karyawan dan aktivitas mereka. Termasuk kegiatan penggajian di perusahaan dan lain-lain. Basis data adalah kumpulan data yang saling berelasi. Data itu sendiri adalah fakta tentang sesuatu ataupun manusia. Data digambarkan sebagai nilai (angka, string, karakter, atau simbol). Basis data dimaksudkan untuk mengatur data sedemikian rupa sehingga mudah, akurat, dan cepat untuk diakses. Manfaat yang kita dapatkan dari basis data sendiri beragam, diantaranya adalah:

1. Kecepatan dan kemudahan

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah.

2. Kebersamaan pemakai

Sebuah database dapat diakses oleh berbagai pengguna yang memiliki hak akses dan dari banyak aplikasi. Untuk database yang digunakan oleh banyak pengguna dan organisasi, tidak perlu menyimpan di setiap bagian, tetapi cukup dengan satu database untuk digunakan bersama.

3. Pemusatan kontrol data

Karena satu database cukup untuk banyak tujuan pemantauan data juga cukup untuk dilakukan di satu tempat.

4. Efisiensi ruang penyimpanan

Dengan penggunaan bersama kita tidak perlu menyediakan ruang penyimpanan di banyak tempat tetapi hanya perlu menyediakan satu untuk menghemat ruang penyimpanan data organisasi.

5. Keakuratan

Menjalankan aturan secara ketat terkait tipe data domain, data keunikan, data hubungan, data lainnya., dapat meningkatkan keakuratan pemasukan atau penyimpanan data.

6. Ketersediaan

Dengan database kita bisa membackup data, mengkategorikan data apa saja yang masih dibutuhkan dan data apa saja yang perlu dibackup di tempat lain.

7. Keamanan

Seagian besar DBMS dilengkapi dengan fungsi manajemen pengguna dan diberi hak akses yang berbeda tergantung pada pengguna dan lokasinya.

8. Kemudahan dalam pembuatan program aplikasi baru

Dengan database pemuatan aplikasi dapat memanfaatkan kemampuan DBMS, sehingga pengembangan aplikasi tidak perlu berurusan dengan penyimpanan data, tetapi hanya perlu mengelola antarmuka pengguna.

9. Pemakain secara langsung

Basis data mampu menampilkan data langsung menggunakan alat yang disediakan oleh DBMS.

10. Kebebasan data

Jika suatu program dihentikan dan ternyata terjadi perubahan isi atau struktur data. Oleh karena itu dengan database modifikasi ini hanya oleh dilakukan pada tingkat DBMS tanpa membongkar program aplikasi.

11. User view

Basis data memberikan tampilan yang berbeda untuk setiap pengguna.

2.2.3 API DAPODIK

Untuk tujuan integrasi dengan sistem eksternal, sistem Dapodik menyediakan layanan komunikasi dgn memanfaatkan web service API (Application Programming Interface). Dengan memanfaatkan layanan komunikasi ini, sistem lain bisa memanfaatkan data-data yang tersedia dalam Dapodik untuk kepentingannya masing-masing. Walaupun demikian, mengingat Dapodik masih dalam tahap pengumpulan dan verifikasi data, untuk sementara layanan ini hanya bersifat searah dan read-only (baca saja). Ke depan, layanan komunikasi ini akan terus dikembangkan dengan menambah fitur modifikasi (insert, delete, update) dan format komunikasi yang lebih beragam (XML, JSON, dan sebagainya).

Untuk komunikasi yang dilakukan menggunakan API DAPODIK yaitu protokol HTTP dan method GET. Layanan atau sistem membutuhkan satu parameter masukan yang harus berisi nilai benar dan valid, untuk kepentingan otentikasi pengguna layanan. Masukan yang diharapkan bernama "passport" yang harus selalu disertakan setiap kali akan mengakses layanan API Dapodik (mandatory request parameter). Nilai passport yang benar dan harus digunakan dalam request API selanjutnya diperoleh dari proses login dimana nilai tersebut hanya akan valid dan diakui selama sesi login pengguna yang bersangkutan masih aktif. Kemudian format keluaran yang tersedia yaitu INI atau CONF.

Layanan API Dapodik sementara hanya menyediakan data secara spesifik dalam lingkup tertentu. Untuk melakukan sinkronisasi data dalam jumlah besar (bulk), harus dilakukan dgn mekanisme perulangan terhadap layanan2 yang telah disediakan. Misal, untuk melakukan sinkronisasi data siswa dalam satu kota/kab, bisa dilakukan dalam urutan berikut:

1. Ambil daftar sekolah dalam kota/kab yang dimaksud. Akan lebih baik jika proses ini dipilah sejumlah kode jenjang yang dikenali agar tidak terlalu membebani server Dapodik.
2. Lakukan sinkronisasi data sekolah di database lokal berdasarkan hasil kembalian dari langkah [1].
3. Ambil daftar siswa untuk setiap sekolah berdasarkan hasil kembalian dari langkah [1].
4. Lakukan sinkronisasi data siswa di database lokal berdasarkan hasil kembalian dari langkah [3].
5. Ulangi langkah [3] dan [4] hingga data siswa dari sekolah yang diperoleh dari langkah [1] seluruhnya berhasil dilakukan.
6. Ulangi langkah [1] hingga [5] untuk jenjang sekolah yang lain, jika langkah [1] dipilah berdasarkan jenjang (disarankan).

Jika mekanisme diatas dilakukan secara rapi dan tercatat, maka apabila terjadi sesuatu hal yang mengganggu proses sinkronisasi, maka tidak perlu dilakukan sinkronisasi ulang terhadap seluruh data, melainkan cukup diulang dari sejak gangguan terjadi. Hal

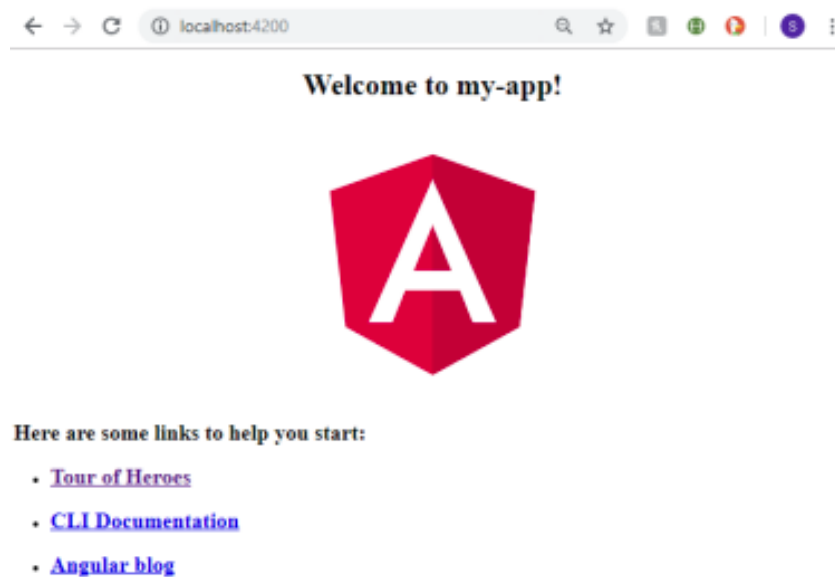
ini memberi keuntungan baik pada sistem Dapodik maupun terhadap sistem lokal yaitu lebih hemat resource terutama penghematan waktu, tenaga mesin, dan bandwidth.

Penggunaan API DAPODIK memerlukan sebuah aktivasi, dimana diperlukan akun API Dapodik yang telah terdaftar dan diaktivasi oleh Admin Dapodik Pusat. Aktivasi akun API Dapodik bisa diajukan melalui aplikasi web operator dapodik atau dengan mengirimkan surat permintaan akun API Dapodik secara resmi dari Sekolah atau Dinas Pendidikan Kota/Kab setempat ke Admin Dapodik Pusat.

2.2.4 Framework Angular

AngularJS adalah framework JavaScript yang dikelola dan dikembangkan oleh Google Inc. dan beberapa kontributor *open source*. Angular awalnya dirilis pada tahun 2009 terutama untuk tujuan memuat sintaks HTML lebih cocok untuk pengembangan aplikasi. Ini termasuk konsep seperti pengikatan data dan template HTML. Dalam bentuknya yang paling sederhana aplikasi AngularJS akan terdiri dari halaman HTML yang disematkan (dengan "variabel") dan objek JavaScript (disebut pengontrol) dengan properti dan metode. Pengembang akan memanipulasi properti utama dan Angular akan secara otomatis memperbarui DOM (halaman HTML) untuk menampilkan perubahan yang dibuat. Angular 2 dirilis pada tahun 2014 dan telah sepenuhnya ditulis ulang untuk memanfaatkan fitur-fitur baru yang memungkinkan pengembangan komponen. Berikut adalah pendekatan lain untuk pengembangan UI; alih-alih membangun halaman dan berharap perpustakaan (*library*) pihak ketiga tidak

bertentangan dengan kode yang dibuat pengembang, pengembang memuat komponen yang akan berfungsi seperti yang diharapkan dan kemudian menggunakan Angular untuk merender komponen bagi pengguna. *Framework Angular* pada saat pertama kali dijalankan akan terdapat tampilan awal aplikasi angular dan terdapat pesan “*Welcome to my-app*”, seperti pada **Gambar 2.3**.



Gambar 2.3 Angular First Run

Angular 2 adalah penulisan ulang lengkap dari perpustakaan Angular dan tidak kompatibel dengan aplikasi Angular 1. Hal ini menimbulkan kekhawatiran di antara pengembang tetapi tim Angular ingin memanfaatkan banyak fitur baru yang tidak tersedia untuk versi sebelumnya. Angular 2 adalah tentang memuat sebagian besar pengembangan browser baru, dapat bergerak maju dan membangun aplikasi yang lebih baik. Terdapat beberapa fitur web baru yang ada di dalam Angular seperti ECMAScript dan TypeScript. ECMAScript adalah struktur bahasa *scripting* berstandar internasional

yang ditetapkan oleh ECMA. JavaScript adalah salah satu implementasi ECMAScript paling populer untuk aplikasi web sisi klien. Ini pertama kali diterbitkan pada tahun 1997 dan telah berkembang selama bertahun-tahun. Versi terbaru (ECMAScript 2018) menambahkan peningkatan sintaks yang signifikan untuk menulis aplikasi skrip yang kompleks. Meskipun tidak semua browser mendukung semua fitur baru, ECMAScript merupakan terobosan terbaru untuk JavaScript. TypeScript adalah superset dari JavaScript yang dikelola dan dikembangkan oleh Microsoft. Itu menambahkan fungsionalitas ke JavaScript terutama tipe data untuk variabel. Pengembang juga menambahkan banyak fungsi dalam ECMAScript (bahasa *scripting* yang didukung oleh sebagian besar *browser* saat ini). Angular sendiri ditulis dalam TypeScript.

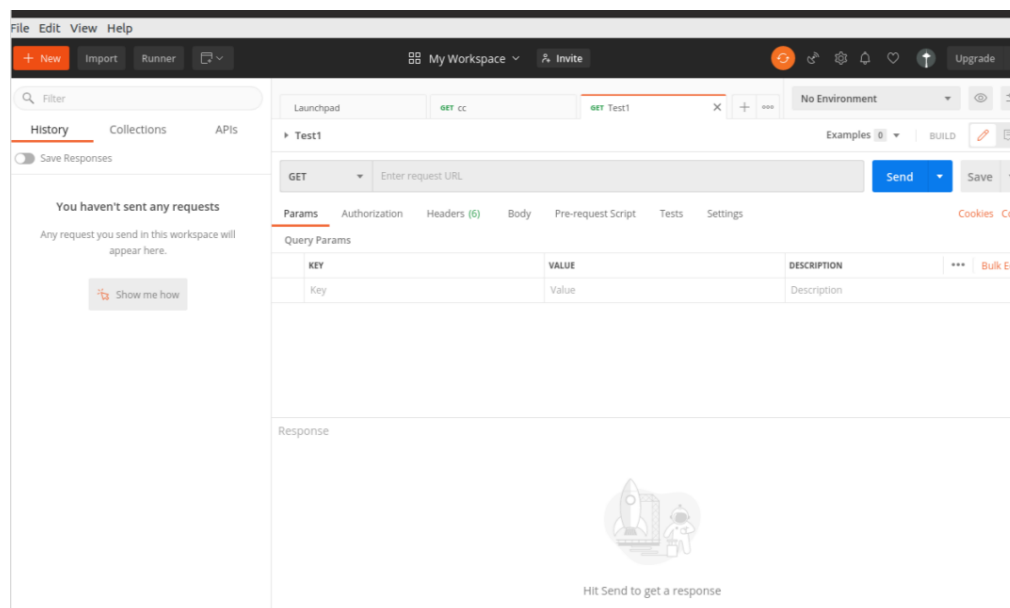
Angular adalah kerangka kerja yang terus berkembang dan meningkat. Versi kerangka kerja saat ini adalah 15.0. Secara umum rilis besar baru direncanakan setiap enam bulan dan rilis baru kompatibel dengan rilis besar terakhir. Jika memungkinkan pengembang harus memperbarui ke rilis utama baru segera setelah stabil. Komponen web pertama kali diperkenalkan pada tahun 2011 meskipun komponen merupakan bagian dari proses pengembangan perangkat lunak selama bertahun-tahun sebelumnya. Standar yang diperlukan untuk mengimplementasikan komponen web sedang dikembangkan oleh W3C dan mewakili masa depan pengemangan aplikasi web.

2.2.5 Postman

Postman adalah platform kolaborasi untuk pengembangan API. Postman dirancang pada tahun 2012 oleh pengembang perangkat lunak dan pengusaha Abhinav Asthana

untuk membuat pengembangan dan pengujian API menjadi mudah. Postman adalah alat untuk menguji perangkat lunak API, yang dapat digunakan untuk merancang, mendokumentasikan, memverifikasi, membuat, dan mengubah API. Fitur Postman yang sederhana membuat pengujian API dapat dilakukan dengan baik dan cepat. Cara kerja Postman dengan mengklasifikasi request berdasarkan request method, URL dan parameter-parameter request (Wagner, 2014).

Postman memiliki fitur untuk mengirim dan mengamati Hypertext Transfer Protocol (HTTP) permintaan dan tanggapan. Postman memiliki antarmuka pengguna grafis (GUI) dan dapat digunakan di platform seperti Linux, Windows dan Mac. Operasi yang dapat dilakukan yaitu membangun beberapa permintaan HTTP – POST, PUT, GET, PATCH dan terjemahkan ke kode. Tampilan dari postman dapat dilihat pada **gambar 2.4**.



Gambar 2.4 Dashboard Postman

Dalam tampilan GUI postman terdapat 4 bagian utama navigasi diantaranya

1. Header

Postman terdiri dari New, Import, Runner (digunakan untuk menjalankan tes dengan Collection Runner), Buka menu Baru, Interceptor, Sync, dan sebagainya. Ini menunjukkan nama ruang kerja bersama dengan opsi Undang untuk membagikannya di antara tim.

2. Response

Bagian respons harus memiliki nilai yang diisi hanya ketika permintaan dibuat. itu umumnya berisi rincian Respon.

3. Sidebar

Sidebar terdiri dari Koleksi (digunakan untuk memelihara tes, berisi folder, sub-folder, permintaan), Riwayat (mencatat semua permintaan API yang dibuat di masa lalu), dan API.



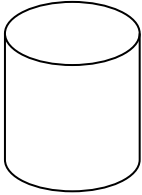
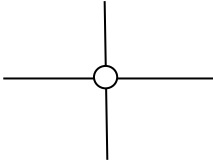
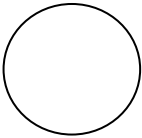
4. Builder

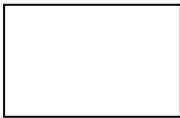
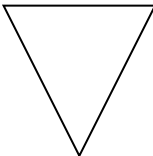
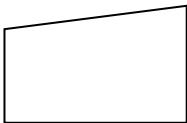

Builder adalah bagian terpenting dari aplikasi Postman. Bagian ini memiliki tab permintaan dan menampilkan nama permintaan saat ini. Secara default, Permintaan Tanpa Judul disebutkan jika tidak ada judul disediakan untuk permintaan.

2.2.6 Flow Map

Class Diagram adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika. Class diagram menggambarkan struktur statis dari sebuah sistem. Simbol dari class diagram dapat dilihat pada **Tabel 2.2**

Tabel 2.2 Simbol Class Diagram

| No | Gambar | Nama | Keterangan |
|----|---|----------------------|---|
| 1 |  | Dokumen | Menunjukkan dokumen berupa dokumen input dan output pada proses manual dan proses berbasis computer |
| 2 |  | Proses Manual | Menunjukkan proses yang dilakukan secara manual |
| 3 |  | Penyimpanan Magnetik | Menunjukkan media penyimpanan data/informasi file pada proses berbasis computer ,file dapat disimpan pada harddisk , disket, CD dan lain-lain |
| 4 |  | Arah Alir Dokumen | Menunjukkan arah aliran dokumen antar bagian yang terkait pada suatu sistem |
| 5 |  | Penghubung | Menunjukkan alir dokumen yang terputus atau terpisah pada halaman alir dokumen yang sama |

| | | | |
|---|---|--------------------|---|
| 6 |  | Proses computer | Menunjukkan proses yang dilakukan secara komputerisasi |
| 7 |  | Pengarsipan | Menunjukkan simpanan data non computer /informasi file pada proses manual. Dokumen dapat disimpan pada lemari , arsip, map file dan |
| 8 |  | Input Keyboard | Menunjukkan input yang dilakukan menggunakan keyboard |
| 9 |  | Penyimpanan manual | Menunjukkan media penyimpanan data atau informasi secara manual |


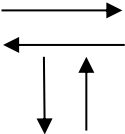
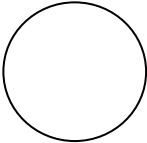
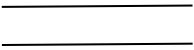
2.2.7 Data Flow Diagram (DFD)

Informasi yang bergerak dalam perangkat lunak mengalami beberapa perubahan. Data flow diagram (DFD) adalah representasi dalam bentuk gambar yang menjelaskan aliran informasi dan perubahan yang terjadi dari suatu data di-input sampai hasil output suatu data. DFD juga dapat digunakan untuk merepresentasikan sistem atau perangkat lunak dalam semua level yang abstrak. (Roger, 2001).

Simbol DFD biasa dipakai dalam berbagai tujuan analisa sistem, termasuk menggambarkan suatu aliran data logis melewati suatu proses. Selain itu DFD dapat memberikan sesuatu yang lebih konseptual, gambaran non-fisik atas pergerakan data

melewati suatu sistem. DFD terdiri dari 4 simbol seperti pada **Tabel 2.3** dibawah ini, yaitu proses, source/sinks, data stores dan data flow lines

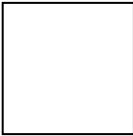
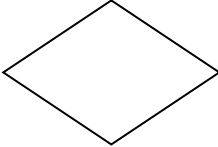
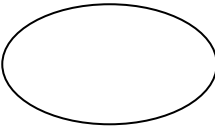

Tabel 2.3 Simbol Data Flow Diagram

| Simbol | Nama | Keterangan |
|---|---------------------------------|---|
|  | Kesatuan Luar (External Entity) | Sesuatu yang berada di luar sistem, tetapi ia memberikan masukan ke dalam sistem atau menerima data dari sistem. External entity tidak termasuk bagian dari sistem. |
|  | Arus Data (Data Flow) | Tempat mengalir informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ini mengalir diantara proses, data store, dan menunjukkan arus data dari data berupa masukan untuk sistem atau hasil proses sistem. |
|  | Proses (Process) | Apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang dihasilkan |
|  | Simpanan Data (Data Store) | Tempat penyimpanan data yang ada dalam sistem, yang disimbolkan dengan sepasang garis sejajar dengan sisi samping terbuka |

2.2.8 Entity Relationship Diagram (ERD)

Entity Relationship Diagram adalah kumpulan konsep yang menguraikan struktur basis data dan suatu hubungan timbal-balik dan proses pembaruan pada basis data. Tujuan utamanya untuk mengembangkan suatu teknik hubungan tingkat tinggi dengan perancangan basis data. ERD digunakan untuk memodelkan 29 struktur data dan hubungan antar data. Dengan ERD, model dapat diuji dengan mengabaikan proses yang dilakukan. Notasi yang digunakan pada Entity Relationship Diagram dapat dilihat pada **Tabel 2.4**. (Supardi, 2010).




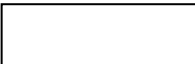
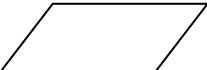
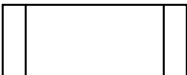
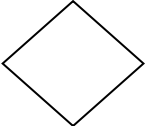
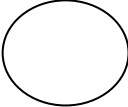
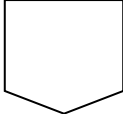
Tabel 2.4 Notasi Entity Relationship Diagram

| Notasi | Keterangan |
|---|--|
|  | Entitas merupakan suatu objek yang dapat diidentifikasi dalam lingkungan pemakai |
|  | Relasi berfungsi sebagai penunjuk adanya hubungan di antara sejumlah entitas yang berbeda |
|  | Atribut merupakan deskripsi karakter entitas |
|  | Garis berfungsi sebagai penghubung antara relasi dengan entitas, relasi dan antitas dengan atribut |

2.2.9 Flowchart

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Berikut pada **Tabel 2.5** adalah beberapa simbol yang digunakan dalam menggambar suatu flowchart.

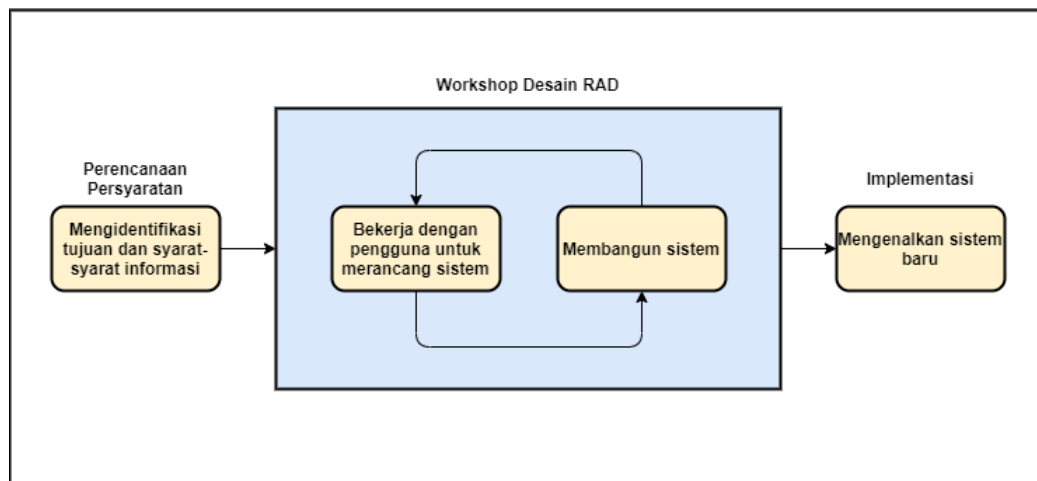
Tabel 2.5 Simbol Flowchart

| Simbol | Nama | Fungsi |
|---|---------------------------------|--|
|  | Terminator | Permulaan atau akhir program |
|  | Garis alir (flow line) | Arah aliran program |
|  | Preparation | Proses inisialisasi |
|  | Proses | Proses perhitungan atau pengelolaan data |
|  | Input/output data | Proses input.output data, parameter, informasi |
|  | Predefined process(sub program) | Permulaan sub program |
|  | Decision | Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk Langkah selanjutnya |
|  | On page connector | Perhitungan bagian flowchart yang berada pada satu halaman |
|  | Off page connector | Penghubung bagian flowchart yang berada pada halaman yang berbeda |

2.2.10 Metode Rapid Application Development (RAD)

RAD adalah sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruksi, cepat, berulang dan bertambah serangkain prototype / prototipe bekerja sebuah sistem yang pada akhirnya berkembang kedalam sistem final (atau sebuah versi) (Whitten, 2004).

Tujuan utama dari semua metode sistem development adalah memberikan suatu sistem yang dapat memenuhi harapan dari para pemakai, tapi terkadang para pemakai tidak dilibatkan langsung dalam melakukan pengembangan sistem sehingga hal ini menyebabkan sistem informasi yang dibuat jauh dari harapan. Metode RAD mempunyai 3 tahapan utama, yaitu :



Gambar 2.5 Metode RAD (Kendall, 2003)

Model pengembangan RAD memiliki empat fase, yaitu fase perencanaan syarat-syarat, fase perancangan, fase konstruksi, dan fase pelaksanaan. Berikut adalah penjelasan fase-fase dalam penelitian ini (Kendall, 2003):

1. Fase perencanaan persyaratan

Pada tahap ini dilakukan pengidentifikasian tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Pada tahap ini penelitian melakukan analisis kebutuhan.

2. Fase perancangan

Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidak sesuaian desain antara user dan analyst. Untuk tahap ini keaktifan user terlibat sangat menentukan untuk mencapai tujuan, karena user bisa langsung memberikan masukan apabila terdapat ketidaksesuaian pada desain.

3. Fase konstruksi

Setelah desain dari sistem yang akan dibuat sudah disetujui baik itu oleh user dan analyst, maka pada tahap ini programmer mengembangkan desain menjadi satu program. Hal terpenting adalah keterlibatan user sangat diperlukan supaya sistem yang dikembangkan dapat sesuai dengan permintaan user.

4. Fase pelaksanaan

Pada tahap ini dilakukan pengujian aplikasi, dengan menggunakan black box testing. Yaitu sebuah pengujian dengan menjalankan setiap fungsi dari aplikasi.

Bab 3

Metodologi

3.1 Subjek Penelitian

Topik dari penelitian ini adalah melakukan pengembangan perangkat lunak pada sebuah sistem berbasis web. Pada penelitian ini sistem yang akan dikembangkan adalah Sistem Informasi untuk Verifikasi dan Validasi Data DAPODIK. Sistem Informasi untuk Verifikasi dan Validasi data DAPODIK digunakan sebagai media pengelolaan data DAPODIK pada sekolah, dimana guru dapat melakukan pengecekan DAPODIK tanpa harus memakai aplikasi DAPODIK dan guru dapat melaporkan data DAPODIK yang keliru, kemudian laporan tersebut diterima oleh operator. Penelitian ini akan menggunakan metode RAD (*Rapid Application Development*) untuk melakukan pengembangan mengikuti tahapan yang ada. Diharapkan hasil penelitian ini akan membuat Sistem Informasi untuk Verifikasi dan Validasi Data DAPODIK yang mampu memenuhi kebutuhan sekolah dalam hal administrasi data DAPODIK.

3.2 Alat Penelitian

Penelitian ini menggunakan *hardware* dan *software*, diantaranya:

3.2.1 Perangkat Keras (Hardware)

Penelitian ini menggunakan sebuah perangkat keras (*Hardware*) berupa laptop dengan spesifikasi seperti pada **Tabel 3.1**.

Tabel 3.1 Spesifikasi Laptop

| No. | Nama | Keterangan |
|-----|-----------|---|
| 1. | Processor | Intel® Core™ i7-8750H (2.20GHz – 4.10GHz) |
| 2. | Graphics | Intel® UHD Graphics 630, Nvidia GeForce GTX 1050/1050 Ti 4GB |
| 3. | Memory | 16GB DDR4 |
| 4. | SSD | Samsung SSD 970 EVO Plus 500GB |
| 5. | Monitor | 15 inci 1920 x 1080 <i>pixel</i> , 300 nits, 144Hz <i>refresh-rate</i> (opsional) |

3.2.2 Perangkat Lunak (Software)

Penelitian ini menggunakan sebuah perangkat lunak (*Software*) yang digunakan untuk pengembangan perangkat lunak, seperti pada **Tabel 3.2**.

Tabel 3.2 Spesifikasi Software Penelitian

| No. | Nama | Keterangan |
|-----|------------------------------|---|
| 1. | Sistem Operasi | Windows 10 Home |
| 2. | Sistem/Aplikasi Objek | Sistem Informasi Inventaris Barang |
| 3. | XAMPP versi 3.3.0 Tahun 2021 | Aplikasi yang berfungsi sebagai server yang independen (localhost) yang didalamnya ada beberapa sistem seperti: Apache, MySQL, FileZilla, Mercury dan Tomcat. |

| | | |
|----|--|--|
| 4. | Visual Studio Code versi 1.61.2 tahun 2021 | Aplikasi yang digunakan sebagai pengolahan kode (kode editor) |
| 5. | MySQL versi 11.11 Tahun 2013 | Aplikasi yang digunakan sebagai pengelolaan database |
| 6. | PhpMyAdmin versi 5.1.1 tahun 2021 | Software yang terdapat didalam XAMPP, berfungsi sebagai pengelolaan database. |
| 7. | Draw.io (desktop) versi 13.9.9 Tahun 2021 | Aplikasi yang digunakan untuk membuat rancangan UI atau prototype |
| 8. | Git versi 2.34.1 Tahun 2021 | Aplikasi yang digunakan untuk melakukan instalasi framework atau hal lainnya yang berasal dari GitHub atau GitLab. |

3.3 Metode Pengembangan Sistem

Adapun dalam tahap pengembangan sistem RAD (*Rapid Application Development*) ini terdiri dari beberapa aktifitas yang tentunya sesuai dengan tahapan yang telah dijabarkan pada alur proses pengembangan sistem. Tahapan tersebut yaitu:

3.3.1 Fase Perencanaan Syarat-Syarat

Menggabungkan laporan hasil metode studi lapangan berupa kebijakan pemakai menjadi spesifikasi yang terstruktur dengan menggunakan pemodelan yang berfungsi untuk mengetahui kebutuhan pemakai, sistem terdahulu dan sistem yang masih manual sehingga data yang ada menjadi tidak teratur. Dari analisis sistem tersebut dapat ditetapkan tujuan perancangan. Tahap yang dilakukan antara lain:

1. Use Case Diagram sistem yang sedang berjalan yang bertujuan untuk lebih mudah mengetahui kekurangan atau kendala sistem yang sedang berjalan.
2. Identifikasi masalah.
3. Pemecahan masalah

3.3.2 Perancangan Sistem (Workshop Design)

Pada tahap ini peneliti mendesain yang diusulkan agar dapat berjalan dengan lebih baik dan diharapkan dapat mengatasi masalah-masalah yang ada.

Penerapan modal yang diinginkan pemakai antara lain:

1. Pada tahap ini alat yang digunakan sama dengan tahap analisis sistem yaitu UML (Unified Modelling Language), alasannya adalah untuk lebih memahami langkah awal membangun sistem secara fisik.
2. Perancangan basis data dilakukan dengan Class Diagram yang menggambarkan hubungan antar entity yang ada pada Use Case Diagram dan spesifikasi tabel.
3. Perancangan Input-Output, dengan membuat rancangan layar tampilan. Setelah rancangan layar tampilan terbentuk maka dilakukan tahap konstruksi.

3.3.3 Implementasi Sistem (Coding & Testing)

Setelah melakukan analisis sistem dan perancangan sistem secara rinci, maka tiba saatnya sistem untuk diimplementasikan. Pada tahap ini terdapat banyak aktivitas yang dilakukan. Aktivitas-aktivitas yang dimaksud berupa:

3.3.3.1 Coding

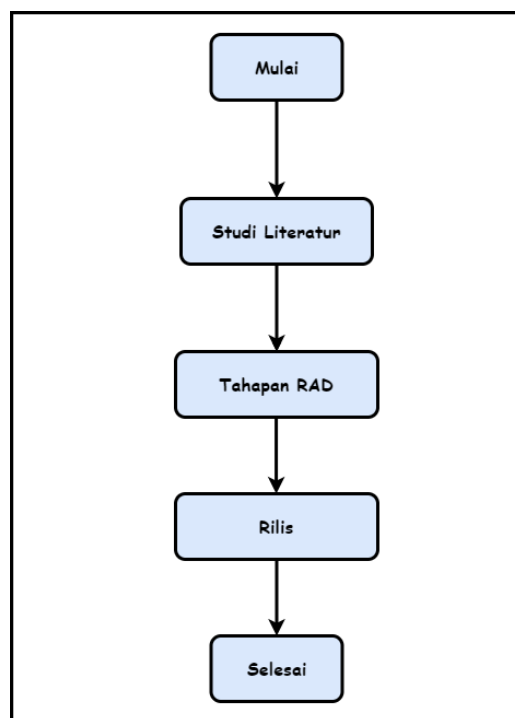
Pada tahap ini hasil desain dimasukkan ke dalam bentuk bahasa pemrograman yang digunakan agar dapat dijalankan dalam sistem atau aplikasi.

3.3.3.2 Testing

Pada tahap ini dilakukan uji coba terhadap sistem yang baru agar dapat digunakan tanpa menemukan kendala-kendala apapun. Adapaun ujicoba yang dilakukan dengan menggunakan metode black box testing.

3.4 Tahap Penelitian

Tahapan penelitian dalam sebuah pengembangan perangkat lunak sangatlah penting, untuk menunjang keberhasilan pengembangan. Sebuah tahapan penelitian tidak selalu sama, tergantung dari apa yang hendak diteliti. Pada pengembangan perangkat lunak tahapan penelitian dapat dilihat pada **Gambar 3.1**.



Gambar 3.1 Tahap Penelitian

Setiap tahapan pada penelitian memiliki perannya masing-masing diantaranya:

1. Studi literatur merupakan proses analisa perbandingan dari penelitian sebelumnya, dengan cara membaca dan mempelajari buku-buku yang berhubungan dengan analisa dan perancangan sistem.
2. Tahapan RAD merupakan tahapan dimana dimulainya penentuan syarat-syarat pembuatan sistem, perancangan sistem seperti pembuatan prototipe dan database, pembuatan sistem (koding), dan tahap pengujian.
3. Rilis merupakan tahap akhir, dimana sistem siap untuk digunakan.

3.5 Jadwal Penelitian

Penelitian ini memiliki beberapa aktifitas yang dilakukan selama 6 bulan, terhitung dari bulan September 2022 – Maret 2023. Aktifitas yang dilakukan seperti pada **Tabel 3.3**.

Tabel 3.3 Jadwal Penelitian

| No | Aktifitas | Bulan | | | | | | |
|----|----------------------------------|----------------|--------------|---------------|---------------|--------------|---------------|------------|
| | | September 2022 | Oktober 2022 | November 2022 | Desember 2022 | Januari 2022 | Februari 2023 | Maret 2023 |
| 1. | Perencanaan Sistem | | | | | | | |
| 2. | Analisis Sistem | | | | | | | |
| 3. | Perancangan dan Pembuatan Sistem | | | | | | | |
| 4. | Pengujian Sistem | | | | | | | |
| 5. | Implementasi Sistem | | | | | | | |

| | | | | | | | | |
|----|---------------------|--|--|--|--|--|--|--|
| 6. | Pemeliharaan Sistem | | | | | | | |
| 7. | Sidang Skripsi | | | | | | | |

Daftar Acuan

- Julio, E., Pakereng, M. A. I., & Artikel, I. (2021). Implementasi API Payment Gateway Menggunakan Arsitektur Microservice. *JURNAL INFORMATIKA*, 8(2). <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- Pranata, B. A., Hijriani Dan, A., Junaidi, A., Komputer, J. I., Matematika, F., Pengetahuan, I., Universitas, A., Jl, L. S., Brodjonegoro, N., & Lampung, B. (2018). *PERANCANGAN APPLICATION PROGRAMMING INTERFACE (API) BERBASIS WEB MENGGUNAKAN GAYA ARSITEKTUR REPRESENTATIONAL STATE TRANSFER (REST) UNTUK PENGEMBANGAN SISTEM INFORMASI ADMINISTRASI PASIEN KLINIK PERAWATAN KULIT* (Vol. 6, Issue 1).
- Putri, M. P., & Effendi, H. (2018). Implementasi Metode Rapid Application Development Pada Website Service Guide “Waterfall Tour South Sumatera.” In *Jurnal SISFOKOM* (Vol. 07).
- Rulloh, A., Mahmudah, D. E., & Kabetta, H. (2017). *Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android*.
- Sembara, S. (2020). *RANCANG BANGUN APPLICATION PROGRAMMING INTERFACE*.
- Abushama, H., M., Alassam, H., A., & Elhaj, F., A. (2020). The effect of test Test-Driven Development and Behavior-Driven Development on Project Success Factors: A Systematic Literature Review Based Study. *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*. IEEE Xplore.
- Beck, K. (2002). *Test-Driven Development By Example*. Addison Wesley. November.
- Buani, D. C. P. (2020). Penerapan Metode Rad Dalam Perancangan (Siremedi) Pada Klinik Halo Fisio. *EVOLUSI : Jurnal Sains Dan Manajemen*, 8(1), 56–62. <https://doi.org/10.31294/evolusi.v8i1.7539>
- Hellmann, T. D., Hosseini-Khayat, A., & Maurer, F. (2010). Supporting test-driven

- development of graphical user interfaces using agile interaction design. *ICSTW 2010 - 3rd International Conference on Software Testing, Verification, and Validation Workshops*, 444–447. <https://doi.org/10.1109/ICSTW.2010.35>
- Jain, P., Sharma, A., & Ahuja, L. (2018). The Impact of Agile Software Development Process on the Quality of Software Product. *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*, 812–815. <https://doi.org/10.1109/ICRITO.2018.8748529>
- Khan, W., Ahmad, W., Luo, B., & Ahmed, E. (2019). SQL database with physical database tuning technique and NoSQL graph database comparisons. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019, Imec*, 110–116. <https://doi.org/10.1109/ITNEC.2019.8729264>
- Komalasari, N., Murad, D. F., Agustine, D., Irsan, M., Budiman, J., & Fernando, E. (2018). Effect of education, performance, position and information technology competency of information systems to performance of information system. *2018 International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2018*, 221–226. <https://doi.org/10.1109/ISRITI.2018.8864437>
- Kosasi, S., & Eka Yuliani, I. D. A. (2015). Penerapan Rapid Application Development Pada Sistem Penjualan Sepeda Online. *Simetris : Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 6(1), 27. <https://doi.org/10.24176/simet.v6i1.234>
- Kyi, K. H., Lu, M., & Li, X. (2016). Development of a user-friendly web-based rainfall-runoff model. *Hydrological Research Letters*, 10(1), 8–14. <https://doi.org/10.3178/hr1.10.8>
- Manuaba, I. B. K. (2019). Combination of test-driven development and behavior-driven development for improving backend testing performance. *Procedia Computer Science*, 157, 79–86. <https://doi.org/10.1016/j.procs.2019.08.144>
- Multazam, M. (2020). Perancangan User Interface dan User Experience pada Placeplus menggunakan pendekatan User Centered Design. *Universitas Islam Indonesia*, 1, 8.
- Nilawati, L., Sulastri, D., & Yuningsih, Y. (2020). Penerapan Model Rapid Application Development Pada Perancangan Sistem Informasi Jasa Pengiriman Barang. *Paradigma - Jurnal Komputer Dan Informatika*, 22(2), 197–204. <https://doi.org/10.31294/p.v22i2.8314>

- Parahita, R., Widiartha, I. B. K., & Zubaidi, A. (2018). Sistem Informasi Perhotelan Berbasis Web Service, Studi Kasus Di Pulau Lombok. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, 1(1), 46.
<https://doi.org/10.29303/jcosine.v1i1.21>
- Purnia, D. S., Rifai, A., & Rahmatullah, S. (2019). Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android. *Seminar Nasional Sains Dan Teknologi 2019*, 1–7.
- Ramirez, M. O. G., De-La-Torre, M., & Monsalve, C. (2019). Methodologies for the design of application frameworks: Systematic review. *2019 8th International Conference on Software Process Improvement, CIMPS 2019 - Applications in Software Engineering*. <https://doi.org/10.1109/CIMPS49236.2019.9082427>
- Sneha, A., & Pushpanjali M., C. (2016). Angular Js. *International Journal of Scientific & Engineering Research*, 7(2), 73–76.
<http://www.ijser.org/researchpaper/Angular-JS.pdf>
- Utomo, S. P., Alfiah, N. H., Sani, Z. A., Hanafi, M., & Primadewi, A. (2020). Informasi Terintegrasi Menggunakan FrameWork CodeIgniter. *Seminar Nasional Dinamika Informatika*, 124–128.