

**PENGEMBANGAN *REST API* PADA  
APLIKASI SAKU (SOLUSI AKUNTANSI UMKM) MENGGUNAKAN  
*FRAMEWORK* LUMEN**

**(Skripsi)**

**Oleh**

**WILDAN MUTAQIN  
NPM 1717051037**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
2022**

**PENGEMBANGAN REST API PADA  
APLIKASI SAKU (SOLUSI AKUNTANSI UMKM) MENGGUNAKAN  
FRAMEWORK LUMEN**

**Oleh**

**WILDAN MUTAQIN**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
SARJANA KOMPUTER**

**Pada**

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2022**

## **ABSTRAK**

### **PENGEMBANGAN REST API PADA APLIKASI SAKU (SOLUSI AKUNTANSI UMKM) MENGGUNAKAN FRAMEWORK LUMEN**

**Oleh**

**Wildan Mutaqin**

Usaha tingkat mikro, kecil, dan menengah atau yang lebih dikenal dengan UMKM, merujuk pada kegiatan usaha yang dilakukan oleh masyarakat umum berdasarkan tingkatan keuntungan atau modal tertentu yang diatur dalam Undang-Undang Republik Indonesia. Seringkali kegiatan UMKM tidak dapat berkembang dengan lebih baik dikarenakan keterbatasan akses masyarakat untuk pembukuan catatan keuangan, yang menyebabkan usaha yang dilakukan tidak berjalan dengan efisien.

Salah satu langkah yang dapat dilakukan untuk meningkatkan pemerataan akses bagi masyarakat yang menjalankan kegiatan UMKM adalah melalui pengembangan sistem informasi akuntansi. Keberagaman akses yang dapat diberikan sistem informasi akuntansi baik melalui *website*, atau perangkat *mobile* dapat menjadi alat pendukung kegiatan UMKM dalam berbagai skala usaha. Dalam rangka mendukung interoperabilitas sistem informasi akuntansi yang dibangun, dalam hal ini adalah aplikasi Solusi Akuntansi UMKM (SAKU), maka perlu dikembangkan *application programming interface* (API) berbasis *representational state transfer* (REST) untuk mendukung proses pertukaran dan sinkronisasi data.

Penelitian ini akan berfokus untuk mengembangkan API yang dapat menjadi jembatan pertukaran dan sinkronisasi data menggunakan metode pengembangan *waterfall* dan *framework* Lumen. Pengembangan API dalam penelitian ini difokuskan untuk menjembatani sistem informasi akuntansi dalam bentuk *website* dan *mobile*, dengan tiga level pengguna yakni, *super-admin*, *admin*, dan *user*.

Kata Kunci : Sistem Informasi Akuntansi, *Representational State Transfer*, *Application Programming Interface*, Lumen.

## **ABSTRACT**

### **REST API DEVELOPMENT ON SAKU APPLICATIONS (SME ACCOUNTING SOLUTIONS) USING LUMEN FRAMEWORK**

**By**

**Wildan Mutaqin**

Micro, small, and medium-sized enterprises or better known as MSMEs, refer to business activities carried out by the general public based on certain levels of profit or capital as regulated in the Law of the Republic of Indonesia. Often MSME activities cannot develop better due to limited public access to financial bookkeeping, which causes the business to be carried out not running efficiently.

One of the steps that can be taken to improve equitable access for people who carry out MSME activities is through the development of an accounting information system. The diversity of access that can be provided by an accounting information system either through a website or mobile device can be a tool to support MSME activities on various business scales. To support the interoperability of the accounting information system that was built, in this case, the Solusi Akuntansi UMKM (SAKU) application, it is necessary to develop an application programming interface (API) based on Representational State Transfer (REST) to support the process of exchanging and synchronizing data.

This research will focus on developing an API that can be a bridge for data exchange and synchronization using the waterfall development method and Lumen framework. The development of the API in this research is focused on bridging the accounting information system in the form of a website and mobile, with three user levels, namely, super-admin, admin, and user.

Keywords: Accounting Information System, *Representational State Transfer*, *Application Programming Interface*, Lumen.

Judul Skripsi : **PENGEMBANGAN REST API PADA  
APLIKASI SAKU (SOLUSI AKUNTANSI  
UMKM) MENGGUNAKAN  
FRAMEWORK LUMEN**

Nama Mahasiswa : **Wildan Mutaqin**

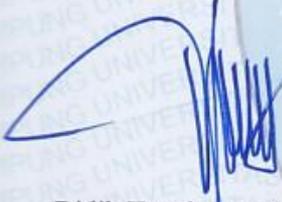
Nomor Pokok Mahasiswa : 1717051037

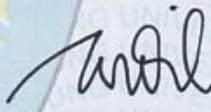
Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

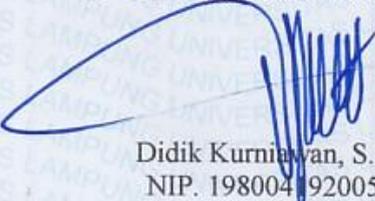
**MENYETUJUI**

1. Komisi Pembimbing

  
Didik Kurniawan, S.Si., M.T.  
NIP. 198004192005011004

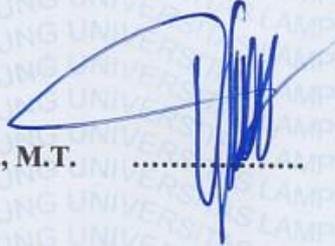
  
Ardiansyah, S.Kom., M.Kom.  
NIP. 198701282018031001

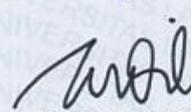
2. Ketua Jurusan Ilmu Komputer

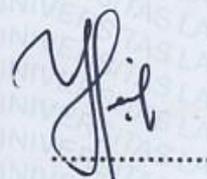
  
Didik Kurniawan, S.Si., M.T.  
NIP. 198004192005011004

**MENGESAHKAN**

1. Tim Penguji

Ketua : **Didik Kurniawan, S.Si., M.T.** 

Sekretaris : **Ardiansyah, S.Kom., M.Kom.** 

Penguji Utama : **Anie Rose Irawati, ST, M.Cs.** 



2. **Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam**



**Prof. Eng. Satripto Dwi Yuwono, S.Si., M.T.**  
NIP 197407052000031001 

**Tanggal Lulus Ujian Skripsi : 31 Desember 2021**

## SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Wildan Mutaqin

NPM : 1717051037

Dengan ini menyatakan bahwa skripsi saya dengan judul “PENGEMBANGAN REST API PADA APLIKASI SAKU (SOLUSI AKUNTANSI UMKM) MENGGUNAKAN FRAMEWORK LUMEN” adalah benar hasil karya sendiri dan bukan orang lain. Seluruh tulisan yang tertuang di dalam skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Jika di kemudian hari terbukti skripsi saya adalah hasil penjiplakan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Bandar Lampung, 28 Mei 2022

Penulis



Wildan Mutaqin  
NPM. 1717051037

## RIWAYAT HIDUP



Penulis dilahirkan di Tangerang pada tanggal 14 Desember 1998 sebagai anak pertama dari dua bersaudara dari pasangan Bapak Maman dan Ibu Tri Susilowati. Penulis menyelesaikan pendidikan Sekolah Dasar (SD) di SDN 4 Tangerang pada tahun 2011. Kemudian menyelesaikan ke jenjang pendidikan menengah pertama di Sekolah Menengah Pertama (SMP) Negeri 2 Tangerang yang diselesaikan pada tahun 2014. Kemudian melanjutkan pendidikan menengah atas di Sekolah Menengah Atas SMA Negeri 4 Tangerang yang diselesaikan pada tahun 2017.

Penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2017 melalui jalur Seleksi Bersama Masuk Perguruan Tinggi (SBMPTN). Selama menjadi mahasiswa, penulis mengikuti beberapa kegiatan antara lain.

1. Anggota Muda Jurusan Ilmu Komputer Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2017/2018.
2. Anggota Bidang Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2017/2018.
3. Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah dasar-dasar pemrograman pada semester ganjil tahun ajaran 2018/2019.
4. Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah sistem operasi pada semester genap tahun ajaran 2018/2019.

5. Praktek Kerja Lapangan (PKL) sebagai magang Sistem Informasi di kantor pusat PT. Perkebunan Nusantara VII.
6. KKN periode ke-II tahun ajaran 2020/2021 di desa Sindang Panon, Kabupaten Tangerang.

## **MOTTO**

**“The best revenge is to improve yourself.” – Ali Bin Abi Thalib**

**“Stay hungry. Stay foolish”- Steve Jobs**

**Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya – (QS. Al-Baqarah: 286)**

*Ku Persembahkan untuk  
Kedua Orang Tua Tercinta*

## SANWACANA

Puji syukur penulis senantiasa terucap atas kehadiran Allah SWT, karena telah memberikan rahmat dan hidayah-Nya kepada saya sehingga saya dapat menyelesaikan skripsi dengan judul “ **Pengembangan REST API Pada Aplikasi SAKU (Solusi Akuntansi UMKM) Menggunakan Framework Lumen**”. Shalawat serta salam penulis sanjungkan kepada junjungan besar Nabi Muhammad SAW yang senantiasa kita bersama nantikan syafaatnya di yaumul akhir.

Penyelesaian skripsi ini tidak terlepas dari adanya bimbingan, kritik, bantuan, semangat dan motivasi yang tiada henti selama penelitian, penulisan serta dalam penyelesaian studi. Saya ucapkan terimakasih kepada :

1. Kedua orang tua saya, Bapak Maman, Ibu Tri Susilowati dan adik saya Hikmal Akbar, yang telah memberikan doa, dukungan, serta motivasi kepada penulis;
2. Bapak Didik Kurniawan, S.Si., M.T. selaku Ketua Jurusan Ilmu Komputer dan pembimbing utama yang telah membimbing, memberikan kritik dan saran, sehingga skripsi ini dapat selesai dengan baik;
3. Bapak Ardiansyah, S.Kom., M.Kom sebagai pembimbing kedua yang telah membimbing dan memberikan ilmu yang bermanfaat;
4. Ibu Anie Rose Irawati, S.T., M.Cs. sebagai pembahas yang telah memberikan masukan yang bermanfaat untuk penyelesaian skripsi ini;
5. Bapak Dr. Eng. Suropto Dwi Yuwono, S.Si., M.T selaku Dekan FMIPA Universitas Lampung;
6. Ibu Astria Hijriani, S.Kom., M.Kom. selaku Sekretaris Jurusan Ilmu Komputer Universitas Lampung;
7. Bapak Ir. Machudor Yusman, M. Kom. selaku Pembimbing Akademik saya di Jurusan Ilmu Komputer Universitas Lampung;

8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu selama menempuh perkuliahan di Jurusan Ilmu Komputer;
9. Ibu Ade Nora Maela, Bang Zainuddin, yang telah membantu urusan administrasi;
10. Alm. Pakde Paing Widodo, Bude Sri Hartini, Mbak Yosi dan Hamik yang telah menjadi keluarga kedua saya di Lampung;
11. Tim Project yang telah berjuang Bersama dalam membuat SAKU, yaitu Brenda Natalia dan Oktaviana;
12. Rekan-Rekan Kumbang Tech(Bonema, Fikri, Patar, Singgih, Adinda, Irham, dan Fauzi) sebagai teman seperjuangan yang mendukung kegiatan akademik dan non-akademik.
13. Rekan dan Senior Himpunan Mahasiswa Jurusan Ilmu Komputer FMIPA Jurusan Ilmu Komputer.
14. Rekan-rekan Ilmu Komputer Angkatan 2017 yang kebersamaian selama masa studi di Jurusan Ilmu Komputer Universitas Lampung.
15. Serta semua pihak yang tidak dapat penulis sebutkan satu-persatu, penulis mengucapkan terimakasih atas doa, dukungan, dan semangat sehingga penulis dapat menyelesaikan skripsi ini.

Akhir kata, penulis menyadari bahwa masih terdapat banyak kekurangan di dalam penyusunan skripsi ini dan jauh dari kesempurnaan, besar harapan penulis, skripsi ini dapat bermanfaat bagi pembaca.

Bandar Lampung, Mei 2022

Penulis,

**Wildan Mutaqin**

## DAFTAR ISI

### Halaman

|   |            |
|---|------------|
| <b>DAFTAR ISI.....</b>                          | <b>i</b>   |
| <b>DAFTAR GAMBAR.....</b>                       | <b>iv</b>  |
| <b>DAFTAR TABEL .....</b>                       | <b>vii</b> |
| <b>I. PENDAHULUAN .....</b>                     | <b>1</b>   |
| 1.1 Latar Belakang.....                         | 1          |
| 1.2 Rumusan Masalah .....                       | 2          |
| 1.3 Batasan Masalah .....                       | 3          |
| 1.4 Tujuan Penelitian .....                     | 3          |
| 1.5 Manfaat Penelitian .....                    | 3          |
| 1.5.1 Bagi Peneliti.....                        | 3          |
| 1.5.2 Bagi Pengguna .....                       | 4          |
| 1.5.3 Bagi Instansi .....                       | 4          |
| <b>II. TINJAUAN PUSTAKA.....</b>                | <b>5</b>   |
| 2.1 Akuntansi Keuangan .....                    | 5          |
| 2.2 Usaha Kecil Menengah dan Mikro (UMKM) ..... | 5          |
| 2.2.1 Usaha Mikro .....                         | 5          |
| 2.2.2 Usaha Kecil.....                          | 6          |
| 2.2.3 Usaha Menengah .....                      | 6          |
| 2.3 REST API.....                               | 6          |
| 2.3.1 Kelebihan .....                           | 6          |
| 2.3.2 Kekurangan.....                           | 7          |

|             |   |           |
|-------------|---|-----------|
| 2.3.3       | Komponen REST API .....                         | 7         |
| 2.4         | JSON ( <i>Javascript Object Notation</i> )..... | 8         |
| 2.5         | Aplikasi.....                                   | 9         |
| 2.6         | <i>Framework Lumen</i> .....                    | 9         |
| 2.7         | PHP.....  | 9         |
| 2.8         | Basis Data.....                                 | 10        |
| 2.9         | Metode <i>Waterfall</i> .....                   | 10        |
| 2.10        | Black-Box Testing.....                          | 12        |
| 2.11        | JSON <i>Web Token</i> (JWT).....                | 12        |
| 2.12        | <i>Unified Modelling Language (UML)</i> .....   | 12        |
| 2.12.1      | <i>Use Case Diagram</i> .....                   | 13        |
| 2.12.2      | Entity Relationship Diagram (ERD).....          | 14        |
| <b>III.</b> | <b>METODOLOGI PENELITIAN .....</b>              | <b>16</b> |
| 3.1         | Waktu dan Tempat Penelitian .....               | 16        |
| 3.2         | Tahapan Penelitian .....                        | 16        |
| 3.2.1       | Studi Literatur .....                           | 17        |
| 3.2.2       | Pengumpulan Data.....                           | 17        |
| 3.2.3       | Pengembangan Sistem.....                        | 17        |
| 3.2.4       | Penelitian Laporan .....                        | 19        |
| 3.3         | Analisa Kebutuhan Sistem .....                  | 19        |
| 3.3.1       | Kebutuhan Fungsional .....                      | 20        |
| 3.3.2       | Kebutuhan <i>Non-Fungsional</i> .....           | 21        |
| 3.4         | Perancangan <i>Database</i> .....               | 22        |
| 3.5         | Skenario Pengujian .....                        | 24        |
| <b>IV.</b>  | <b>HASIL DAN PEMBAHASAN .....</b>               | <b>25</b> |
| 4.1         | Hasil Implementasi.....                         | 25        |

|           |                                |           |
|-----------|--------------------------------|-----------|
| 4.2       | Pengujian Program .....        | 27        |
| 4.2.1     | Pengujian Super-admin.....     | 28        |
| 4.2.2     | Pengujian <i>Admin</i> .....   | 31        |
| 4.2.3     | Pengujian <i>User</i> .....    | 39        |
| 4.2.4     | Pengujian Keamanan .....       | 75        |
| <b>V.</b> | <b>SIMPULAN DAN SARAN.....</b> | <b>78</b> |
| 5.1       | Kesimpulan.....                | 78        |
| 5.2       | Saran .....                    | 79        |
|           | <b>DAFTAR PUSTAKA .....</b>    | <b>81</b> |

## DAFTAR GAMBAR

| Gambar   | Halaman |
|--|---------|
| Gambar 1. Metode <i>Waterfall</i> (Pressman and Maxim, 2020). .....      | 11      |
| Gambar 2. Tahapan Penelitian dan Pengembangan Aplikasi SAKU. ....        | 16      |
| Gambar 3. <i>Use Case Diagram</i> Aplikasi SAKU.....                     | 20      |
| Gambar 4. <i>ERD</i> Aplikasi Solusi Akuntansi UMKM (SAKU).....          | 23      |
| Gambar 5. Pengujian Menampilkan Daftar <i>Admin</i> Aktif.....           | 28      |
| Gambar 6. Pengujian Menambahkan <i>Admin</i> Baru. ....                  | 29      |
| Gambar 7. Pengujian Mengubah Data <i>Admin</i> .....                     | 29      |
| Gambar 8. Pengujian Menghapus <i>Admin</i> .....                         | 30      |
| Gambar 9. Pengujian Menampilkan Pengguna Aktif. ....                     | 31      |
| Gambar 10. Pengujian Merubah Data Pengguna Oleh <i>Admin</i> .....       | 32      |
| Gambar 11. Pengujian Menghapus Akun Pengguna.....                        | 32      |
| Gambar 12. Pengujian Menampilkan Pengguna Tidak Aktif. ....              | 33      |
| Gambar 13. Pengujian Pengaktifan Akun Pengguna Tidak Aktif. ....         | 33      |
| Gambar 14. Pengujian Menampilkan Permintaan Pengaktifan Akun Kembali.... | 34      |
| Gambar 15. Pengujian Mengaktifkan Permintaan Pengembalian Akun.....      | 34      |
| Gambar 16. Pengujian Menandai Pesan Telah Dibaca. ....                   | 35      |
| Gambar 17. Pengujian Menampilkan Master Data Perkiraan.....              | 36      |
| Gambar 18. Pengujian Penambahan Master Data Perkiraan. ....              | 36      |
| Gambar 19. Pengujian Memperbarui Master Data Perkiraan. ....             | 37      |
| Gambar 20. Pengujian Penghapusan Master Data Perkiraan.....              | 37      |
| Gambar 21. Pengujian Masuk Aplikasi. ....                                | 40      |
| Gambar 22. Pengujian Masuk Aplikasi Menggunakan Akun <i>Google</i> ..... | 41      |
| Gambar 23. Pengujian Permintaan Lupa <i>Password</i> .....               | 41      |
| Gambar 24. Pengujian Verifikasi Lupa <i>Password</i> .....               | 42      |
| Gambar 25. Pengujian Reset <i>Password</i> . ....                        | 43      |
| Gambar 26. Pengujian Mengganti <i>Password</i> .....                     | 43      |

|   |    |
|---|----|
| Gambar 27. Pengujian Merubah Data <i>User</i> . .....             | 44 |
| Gambar 28. Pengujian Permintaan Mengaktifkan Kembali Akun.....    | 44 |
| Gambar 29. Pengujian Verifikasi OTP Pengaktifan Akun Kembali..... | 45 |
| Gambar 30. Pengujian Mengambil Data Rekening.....                 | 46 |
| Gambar 31. Pengujian Mengambil Data Perkiraan.....                | 46 |
| Gambar 32. Pengujian Menambah Data Perkiraan. ....                | 47 |
| Gambar 33. Pengujian Mengubah Data Perkiraan.....                 | 48 |
| Gambar 34. Pengujian Menghapus Data Perkiraan. ....               | 48 |
| Gambar 35. Pengujian Mengambil Data Jenis Transaksi. ....         | 49 |
| Gambar 36. Pengujian Mengambil Data Mapping Debit.....            | 49 |
| Gambar 37. Pengujian Mengambil Data Mapping Credit. ....          | 50 |
| Gambar 38. Pengujian Menambah Jurnal Otomatis. ....               | 51 |
| Gambar 39. Pengujian Menambah Jurnal Manual. ....                 | 51 |
| Gambar 40. Pengujian Mengubah Jurnal Otomatis. ....               | 52 |
| Gambar 41. Pengujian Mengubah Jurnal Manual.....                  | 53 |
| Gambar 42. Pengujian Menampilkan List Jurnal.....                 | 53 |
| Gambar 43. Pengujian Menampilkan Detail Jurnal. ....              | 54 |
| Gambar 44. Pengujian Mengambil Pemasukan Pengeluaran. ....        | 55 |
| Gambar 45. Pengujian Menghapus Data Jurnal. ....                  | 55 |
| Gambar 46. Pengujian Laporan Jurnal pada JSON Android. ....       | 56 |
| Gambar 47. Pengujian Laporan Buku Besar.....                      | 56 |
| Gambar 48. Pengujian Laporan Neraca Saldo. ....                   | 57 |
| Gambar 49. Pengujian Laporan Laba Rugi.....                       | 58 |
| Gambar 50. Pengujian Neraca JSON Android.....                     | 59 |
| Gambar 51. Pengujian Perubahan Saldo JSON Android. ....           | 59 |
| Gambar 52. Pengujian Unduh Jurnal Umum. ....                      | 60 |
| Gambar 53. Pengujian Unduh Buku Besar. ....                       | 61 |
| Gambar 54. Pengujian Unduh Neraca Saldo.....                      | 61 |
| Gambar 55. Pengujian Unduh Laba Rugi. ....                        | 62 |
| Gambar 56. Pengujian Neraca.....                                  | 63 |
| Gambar 57. Pengujian Perubahan Saldo.....                         | 63 |
| Gambar 58. Pengujian Ambil Data Hutang Piutang. ....              | 64 |

|  |    |
|--|----|
| Gambar 59. Pengujian Ambil Data Detail Hutang Piutang. ....          | 65 |
| Gambar 60. Pengujian Ambil Data Mapping Bayar Hutang / Piutang. .... | 65 |
| Gambar 61. Pengujian Bayar Hutang Piutang. ....                      | 66 |
| Gambar 62. Pengujian Hapus Data Hutang Piutang. ....                 | 66 |
| Gambar 63. Pengujian Daftar. ....                                    | 67 |
| Gambar 64. Pengujian Hapus Akun. ....                                | 67 |
| Gambar 65. Pengujian Validasi OTP Email. ....                        | 68 |
| Gambar 66. Pengujian Sesi Aktif Pengguna. ....                       | 68 |
| Gambar 67. Pengujian Keluar. ....                                    | 69 |
| Gambar 68. Contoh Bentuk JSON <i>Web Token</i> (JWT). ....           | 75 |
| Gambar 69. Header pada JSON <i>Web Token</i> (JWT). ....             | 75 |
| Gambar 70. Payload Pada JSON <i>Web Token</i> (JWT). ....            | 75 |
| Gambar 71. Signature Pada JSON <i>Web Token</i> (JWT). ....          | 76 |
| Gambar 72. Login Akun. ....  | 76 |
| Gambar 73. Hasil Uji Coba. ....                                      | 77 |

**DAFTAR TABEL**

| Tabel   | Halaman |
|---|---------|
| Tabel 1 Elemen <i>Use Case Diagram</i> .....  | 13      |
| Tabel 2 Elemen <i>Entity Relationship Diagram</i> .....                                 | 15      |
| Tabel 3 Contoh Skenario Pengujian <i>Equivalence Partitioning</i> Pada <i>API</i> ..... | 24      |
| Tabel 4. Implementasi Kebutuhan Fungsional API SAKU .....                               | 25      |
| Tabel 5. Daftar <i>Entity</i> Aplikasi SAKU .....                                       | 26      |
| Tabel 6. Hasil Pengujian <i>Equivalence Partitioning Super-admin</i> .....              | 30      |
| Tabel 7. Hasil Pengujian <i>Equivalence Partitioning Admin</i> .....                    | 38      |
| Tabel 8. Hasil Pengujian <i>Equivalence Partitioning User</i> .....                     | 69      |

## I. PENDAHULUAN

### 1.1 Latar Belakang

Istilah UMKM merujuk pada aktivitas usaha yang didirikan oleh masyarakat, baik berbentuk usaha perseorangan maupun badan usaha. UMKM sendiri merupakan singkatan dari Usaha Mikro, Kecil, dan Menengah. Pada UMKM, pengelolaan/kepemimpinan biasanya dilakukan oleh pemiliknya sehingga terjadi perangkapan jabatan. Pemilik bertindak sebagai pengelola utama usaha sekaligus manajer produksi ataupun sebagai manajer keuangan.

Di dalam sebuah usaha terdapat bagian yang penting untuk dilakukan yaitu melakukan pencatatan arus keuangan. Namun, di dalam praktiknya UMKM seringkali terhambat dalam proses pencatatan arus keuangan dikarenakan kurang pahamiannya pemilik UMKM terhadap aturan-aturan dalam melakukan pencatatan pembukuan keuangan. Sistem informasi akuntansi menjadi salah satu hal penting yang dapat mendukung kelancaran dalam kegiatan operasional perusahaan, baik itu usaha besar ataupun usaha kecil.

Sistem informasi memiliki beberapa *platform* diantaranya *desktop*, *web*, dan *mobile*. Agar sebuah aplikasi dapat berjalan di *platform* berbeda dengan menggunakan *database* yang sama, dibutuhkan jembatan yang dapat menghubungkan perbedaan *platform* tersebut. Salah satu jembatan yang dapat menghubungkannya adalah API (*Application Programming Interface*). REST API adalah sebuah implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data (Perdana, 2018). Seiring berkembangnya internet, protokol HTTP menjadi hal umum yang digunakan untuk transfer data.

Saat ini, pengembangan REST API dapat dilakukan dengan menggunakan *native* PHP maupun *framework*. Dalam pengembangannya, *native* PHP memerlukan pembuatan komponen dari awal, sedangkan *framework* sudah menyediakan komponen yang dapat digunakan oleh pengembang, sehingga pengembangan REST API menggunakan *native* PHP akan memerlukan waktu yang cukup lama jika dibandingkan dengan *framework*.

Disini pengembang akan menggunakan *framework* Lumen dalam pengembangan REST API karena *framework* Lumen mempunyai skor *benchmark* yang lebih tinggi dibanding *framework* lain. Selain itu *framework* Lumen memiliki komunitas terbesar karena Lumen berbasis Laravel sehingga penyelesaian masalah pada saat pengembangan REST API akan dapat lebih mudah dilakukan.

Berdasarkan uraian di atas, maka pada penelitian ini akan dikembangkan REST API untuk aplikasi SAKU (Solusi Akuntansi UMKM). Hasil penelitian ini diharapkan menjadi sebuah layanan untuk komunikasi antar *server* dengan *client* yang akan digunakan pada *frontend* aplikasi SAKU berbasis *Android* dan *Web*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang dapat dirumuskan pada penelitian ini adalah bagaimana mengembangkan sebuah REST API yang dapat menjadi sebuah *backend* untuk aplikasi SAKU baik *mobile* ataupun *web*.

### 1.3 Batasan Masalah

Untuk mengatasi permasalahan penelitian tidak meluas, maka peneliti membuat batasan-batasan masalah sebagai berikut :

1. API dikembangkan menggunakan bahasa pemrograman PHP, *Framework* Lumen dan menggunakan basis data MySQL.
2. API yang dikembangkan, dikhususkan untuk pengembang internal aplikasi SAKU pada *platform web* dan *mobile*.
3. Format keluaran dari API adalah *Javascript Object Notation* (JSON).
4. API yang dikembangkan pada aplikasi SAKU mencakup fitur memasukkan data transaksi, data perusahaan, melihat laporan jurnal, buku besar, neraca saldo, neraca, perubahan modal dan laba rugi, serta melihat detail perusahaan.

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menghasilkan sebuah API akuntansi yang dapat menjadi *backend* pada aplikasi SAKU versi *mobile* ataupun *web*.

### 1.5 Manfaat Penelitian

Manfaat pelaksanaan penelitian ini dijabarkan sebagai berikut :

#### 1.5.1 Bagi Peneliti

1. Menambah wawasan dan pengetahuan terutama di dalam pengembangan API.
2. Menambah wawasan dan pengetahuan tentang akuntansi.

### **1.5.2 Bagi Pengguna**

1. Mempermudah *user* dalam mendapatkan data di dalam penjualan seperti melihat laporan jurnal, buku besar, neraca saldo, neraca, perubahan modal dan laba rugi serta melihat detail perusahaan.
2. Kebebasan dalam memilih Bahasa pemrograman, karena kebanyakan Bahasa pemrograman saat ini sudah mendukung untuk mengkonsumsi data JSON melalui REST API.

### **1.5.3 Bagi Instansi**

1. Menambah perbendaharaan di Jurusan Ilmu Komputer.
2. Sebagai referensi untuk teman-teman dan adik tingkat jika ingin melakukan pengembangan lebih lanjut.

## II. TINJAUAN PUSTAKA

### 2.1 Akuntansi Keuangan

Akuntansi keuangan berorientasi pada pelaporan pihak eksternal. Beragamnya pihak eksternal dengan tujuan spesifik bagi masing-masing pihak membuat pihak penyusun laporan keuangan menggunakan prinsip dan asumsi-asumsi dalam penyusunan laporan keuangan. Untuk itu diperlukan standar akuntansi yang dijadikan pedoman baik oleh penyusun maupun oleh pembaca laporan keuangan. Laporan yang dihasilkan dari akuntansi keuangan berupa laporan keuangan untuk tujuan umum (*general purpose financial statement*) (Martani dkk., 2016).

### 2.2 Usaha Kecil Menengah dan Mikro (UMKM)

UMKM adalah usaha perdagangan yang dikelola oleh badan usaha atau perorangan yang merujuk pada usaha ekonomi produktif sesuai dengan kriteria yang ditetapkan oleh Undang-Undang Nomor 20 Tahun 2008 (*Undang-Undang Republik Indonesia Nomor 20 Tentang Usaha Mikro, Kecil, Dan Menengah*, 2008).

#### 2.2.1 Usaha Mikro

Usaha produktif milik perseorangan dan/atau badan usaha perseorangan yang memenuhi kriteria usaha mikro yang diatur dalam Undang-Undang.

### 2.2.2 Usaha Kecil

Usaha ekonomi produktif yang berdiri sendiri dan dilakukan oleh perseorangan atau badan usaha yang bukan merupakan anak perusahaan atau bukan cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dari usaha menengah atau usaha besar yang memenuhi kriteria usaha kecil yang diatur dalam undang-undang.

### 2.2.3 Usaha Menengah

Usaha ekonomi produktif yang berdiri sendiri, yang dilakukan oleh perseorangan atau badan usaha yang bukan merupakan anak perusahaan atau cabang perusahaan yang dimiliki, dikuasai, atau menjadi bagian baik langsung maupun tidak langsung dengan usaha kecil atau usaha besar dengan jumlah kekayaan bersih atau hasil penjualan tahunan sebagaimana diatur dalam undang-undang.

## 2.3 REST API

REST API adalah sebuah implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data (Perdana, 2018).

### 2.3.1 Kelebihan

Menurut Halili dan Ramadani (2018), kelebihan REST API adalah :

1. Mendukung komunikasi *stateless*.
2. *Bandwidth* yang ringan, dikarenakan menggunakan format pesan JSON.
3. Dapat digunakan oleh berbagai *client*.

### 2.3.2 Kekurangan

Menurut Halili dan Ramadani (2018), berikut ini kekurangan dari REST API :

1. Tidak cocok untuk data dalam jumlah besar.
2. *Request* REST (terutama *GET*) tidak cocok untuk data dalam jumlah besar.
3. Latensi dalam waktu pemrosesan permintaan dan penggunaan *bandwidth*.

### 2.3.3 Komponen REST API

Komponen pembentuk REST API dijabarkan sebagai berikut :

#### 2.3.3.1 URL *Design*

REST API diakses menggunakan protokol HTTP, oleh karena itu diperlukan penamaan dan struktur URL yang baik dan mudah dimengerti dalam penggunaannya. URL API biasa disebut sebagai *endpoint* dalam pemanggilannya. Contoh pemanggilan URL yang baik adalah seperti berikut: *users*, *users/123*, *users/123/photos* dan seterusnya.

#### 2.3.3.2 HTTP *verbs*

HTTP *verbs* adalah sebutan metode yang dilakukan ketika melakukan *request* sehingga *server* mengetahui apa yang ingin *client* dapatkan. Metode ini ada beberapa macam namun yang paling sering dipakai adalah: *GET*, *POST*, *PUT* dan *DELETE*.

### 2.3.3.3 HTTP response code

HTTP *response code* adalah kode yang telah menjadi standar dalam menginformasikan hasil *request* kepada *client*. Secara umum terdapat 3 kelompok kode yang paling sering digunakan di REST API yaitu:

1. 2XX, yang menandakan *request* yang dilakukan berhasil.
2. 4XX, yang menandakan bahwa *request* mengalami kesalahan pada sisi *client*.
3. 5XX, yang menandakan bahwa *request* mengalami kesalahan pada sisi server.

### 2.3.3.4 Format Response

Setiap *request* yang dilakukan *client* akan menerima data *response* dari server, *response* tersebut berupa data XML ataupun JSON, untuk kemudian data tersebut dapat diuraikan (*parsing*) sesuai dengan kebutuhan *client*.

## 2.4 JSON (*Javascript Object Notation*)

JSON (*Javascript Object Notation*) merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan. Selain JSON, terdapat XML yang memiliki fungsi sama yaitu alat komunikasi antar aplikasi, integrasi data, dan komunikasi aplikasi eksternal dengan luaran. JSON lebih unggul dari XML, mulai dari kecepatan, penelitian yang lebih mudah dan *coding* untuk *parsing* yang lebih ringkas dan sederhana (Qibtiyah dan Rahayu, 2017).

## 2.5 Aplikasi

Menurut Abdurahman dan Riswaya (2014), aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan.

## 2.6 *Framework* Lumen

Lumen merupakan salah satu *micro framework* yang dikembangkan oleh Taylor Otwell, pengembang yang berada di balik *framework* PHP Laravel. Lumen lebih dikhususkan untuk pembuatan API, yang mengakibatkan beberapa fitur yang ada di Laravel dipangkas dan tidak tersedia di Lumen. Namun demikian, Lumen memiliki performa yang lebih baik dalam menangani permintaan (*request*) dibandingkan Laravel (Surahman dkk., 2018).

## 2.7 PHP

PHP atau singkatan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *web* dan dapat ditanamkan pada sebuah *script* HTML. Bahasa PHP dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, Java, dan Perl serta mudah untuk dipelajari.

PHP merupakan bahasa *scripting server – side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Secara sederhana, *server* akan menerjemahkan *script* program, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan.

Adapun pengertian lain, yaitu suatu bahasa pemrograman berbasis kode – kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML. (Firman dkk., 2016).

## 2.8 Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan dalam perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Basis data merupakan salah satu komponen utama dalam sistem informasi, karena merupakan basis dalam penyediaan informasi bagi para pemakai.

Pengelolaan basis data secara spesifik tidak dilakukan oleh pemakai secara langsung, tetapi ditangani oleh sebuah perangkat lunak yang khusus (spesifik). Perangkat lunak inilah yang disebut DBMS (*Database Management System*) yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. Ia juga menerapkan mekanisme pengaman data, pemakai data secara bersama, pemaksaan keakuratan atau konsistensi data dan sebagainya.

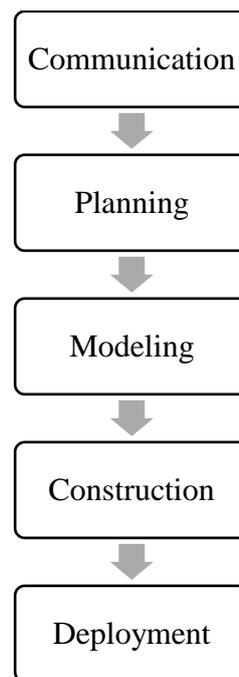
Aplikasi-aplikasi tambahan bersifat opsional (bisa ada dan bisa tidak) dan biasanya terdapat pada DBMS sebagai fungsi tambahan. Ada puluhan bahkan mungkin ratusan perangkat lunak DBMS yang tersedia. Masing-masing dengan spesifikasi sendiri-sendiri. Mulai dari yang sangat sederhana sampai yang paling kompleks, perangkat lunak DBMS itu salah satunya yaitu MySQL (Aunur, 2018).

## 2.9 Metode *Waterfall*

Metodologi *waterfall* adalah metode pengembangan sistem yang dilakukan secara berurutan dari fase ke fase. Menurut Pressman and Maxim (2020),

metode *waterfall* memiliki 5 fase pengembangan yaitu *Communication*, *Planning*, *Modelling*, *Construction*, dan *Deployment*.

Dalam penggunaan metode *waterfall* setiap tahapan akan dilakukan secara terurut, sehingga setiap fase tidak dapat saling tumpang tindih. Hal ini memberikan keuntungan berupa penjadwalan yang mudah untuk setiap fase pengembangan. Namun, Hal ini menyebabkan metode ini tidak terlalu baik untuk digunakan dalam pengembangan perangkat lunak yang memiliki kompleksitas pengembangan yang tinggi. Selain itu, apabila salah satu fase pengembangan tertunda, maka fase selanjutnya tidak bisa dilakukan dan akan memperpanjang tahapan pengembangan. Adapun alur setiap fase pengembangan pada metode *waterfall* disajikan dalam Gambar 1.



Gambar 1. Metode *Waterfall* (Pressman and Maxim, 2020).

Gambar 1, menunjukkan alur fase pengembangan dalam metode pengembangan perangkat lunak *waterfall*.

## 2.10 Black-Box Testing

Menurut Latif (2015), metode *Black box Testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Proses *Black box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan (Wahyudi dkk., 2016).

## 2.11 JSON Web Token (JWT)

JSON Web Token (JWT) merupakan sebuah token berbentuk string JSON yang sangat padat (ukurannya), informasi mandiri yang gunanya sendiri untuk melakukan sistem autentikasi dan pertukaran informasi. Token JWT memiliki bentuk yang kecil, sehingga dapat dikirim melalui URL, parameter HTTP POST atau dalam Header HTTP dan dapat ditransmisikan dengan lebih cepat.

Disebut informasi mandiri karena isi dari token yang dihasilkan memiliki informasi dari pengguna yang dibutuhkan, sehingga tidak perlu *query* ke *database* lebih dari satu kali. Token tersebut dapat diverifikasi dan dipercaya karena sudah terdaftar secara digital. Token JWT dapat dimasukkan dengan menggunakan algoritma *Hash-based Message Authentication Code* (HMAC) (*secret*) atau algoritma RSA (*Rivest-Shamir-Adleman*) (*public/private key*) (Rahmatulloh dkk., 2018).

## 2.12 Unified Modelling Language (UML)

Menurut Putra dan Andriani (2019), UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*,

membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Adapun beberapa diagram yang digunakan dalam perancangan sistem pada penelitian ini yaitu sebagai berikut.

### 2.12.1 Use Case Diagram

Menurut Putra dan Andriani (2019), *use case diagram* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai. Adapun elemen-elemen pembentuk *Use Case Diagram* yang dapat dilihat pada Tabel 1.

Tabel 1 Elemen *Use Case Diagram*

| No | Gambar  | Nama                  | Keterangan  |
|----|---|-----------------------|---|
| 1  |  | <i>Actor</i>          | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i>   |
| 2  |  | <i>Dependency</i>     | Hubungan dimana perubahan yang terjadi pada elemen mandiri (independent) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri (not-independent). |
| 3  |  | <i>Generalization</i> | Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).                 |
| 4  |  | <i>Include</i>        | Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.  |

Tabel 1 (lanjutan)

| No | Gambar  | Nama                 | Keterangan  |
|----|---|----------------------|---|
| 5  |    | <i>Extend</i>        | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.                   |
| 6  |    | <i>Association</i>   | Apa yang menghubungkan antara objek satu dengan objek lainnya.  |
| 7  |    | <i>System</i>        | Menspesifikasikan paket yang menampilkan sistem secara terbatas.  |
| 8  |    | <i>Use Case</i>      | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor                               |
| 9  |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen elemennya (sinergi). |
| 10 |  | <i>Note</i>          | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.  |

### 2.12.2 Entity Relationship Diagram (ERD)

Menurut Fridayanthie dan Mahdiati (2016), ERD (*Entity relationship diagram*) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model. ERD akan sangat membantu dalam proses menerjemahkan kebutuhan *database* untuk sebuah

sistem. Setiap *entity* yang digambarkan dalam ERD mewakili satu entitas yang akan dibentuk dalam *database*. Pembuatan ERD memiliki beberapa aturan penulisan notasi untuk setiap elemen yang akan digunakan, adapun elemen-elemen dari *Entity relationship diagram* disajikan dalam Tabel 2.

Tabel 2 Elemen *Entity Relationship Diagram*

| No | Gambar  | Nama                  | Keterangan   |
|----|---|-----------------------|--|
| 1  |    | <i>Generalization</i> | Hubungan dimana objek anak ( <i>descendent</i> ) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ). |
| 2  |  | <i>Class</i>          | Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.   |

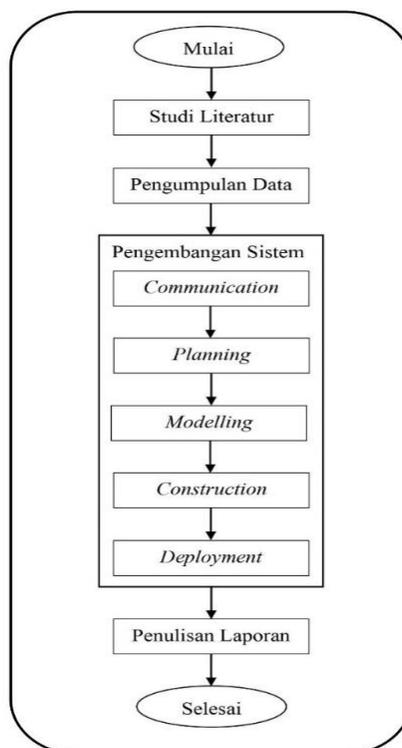
### III. METODOLOGI PENELITIAN

#### 3.1 Waktu dan Tempat Penelitian

Penelitian dilakukan di jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang beralamat di Jalan Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Gedong Meneng, Kecamatan Rajabasa, Kota Bandar Lampung. Penelitian ini dilakukan pada semester genap Tahun Ajaran 2020/2021.

#### 3.2 Tahapan Penelitian

Tahapan penelitian ditunjukkan pada Gambar 2.



Gambar 2. Tahapan Penelitian dan Pengembangan Aplikasi SAKU.

### **3.2.1 Studi Literatur**

Tahapan yang pertama yaitu melakukan studi literatur, yaitu tahapan yang dilakukan agar memahami konsep REST API yang akan dibuat. Dan juga memahami permasalahan yang berkaitan di dalam penelitian ini seperti akuntansi, UMKM dan sebagainya.

### **3.2.2 Pengumpulan Data**

Tahapan selanjutnya melakukan pengumpulan data, di sini cara yang dilakukan ada dua, yaitu observasi dan studi pustaka.

#### **3.2.2.1 Observasi**

Observasi dilakukan untuk membantu dalam pembuatan fitur-fitur dengan cara mempelajari dan mengamati aplikasi yang telah dibuat sebelumnya.

#### **3.2.2.2 Studi Pustaka**

Studi Pustaka yaitu mendapatkan data dari literatur, yang didapatkan dari buku, jurnal maupun internet yang menyajikan tentang data-data yang berkaitan dengan *REST API* ataupun akuntansi UMKM.

### **3.2.3 Pengembangan Sistem**

Tahap selanjutnya yaitu melakukan pengembangan sistem, pada tahap ini menggunakan metode *waterfall*. Mulai dari *communication*, *planning*, *modelling*, *construction* dan *deployment*.

### 3.2.3.1 Communication

Langkah ini merupakan analisis terhadap kebutuhan *software*, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan akuntan ataupun ahli, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

### 3.2.3.2 Planning

Proses *planning* merupakan lanjutan dari proses *communication (analysis requirement)*. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan *software*, termasuk rencana yang akan dilakukan.

### 3.2.3.3 Modeling

Proses *modeling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan *software* yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.

### 3.2.3.4 Construction

*Construction* merupakan proses membuat kode. *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*.

Tahapan ini merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

### **3.2.3.5 Deployment**

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean, sistem yang sudah jadi akan digunakan oleh *user*. Kemudian, pemeliharaan *software* dilakukan secara berkala.

### **3.2.4 Penelitian Laporan**

Yang terakhir adalah pembuatan laporan, yaitu pada tahap ini peneliti menulis ulang kembali langkah-langkah yang dilakukan selama penelitian berlangsung.

## **3.3 Analisa Kebutuhan Sistem**

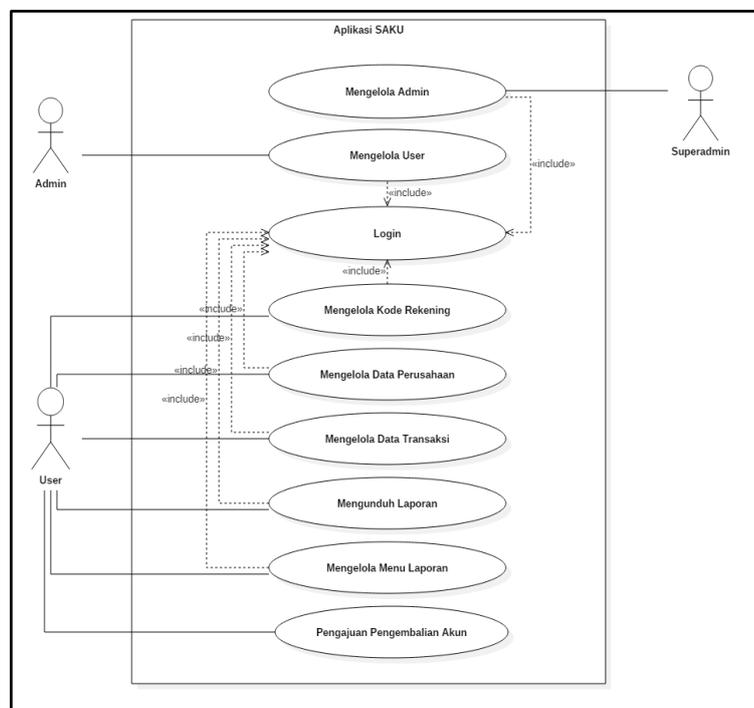
Analisa kebutuhan untuk Pengembangan *REST API* pada Aplikasi SAKU ini dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan *non* fungsional.

### 3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan aplikasi yang berhubungan dengan proses *input* dan *output* pada aplikasi. Berguna untuk mengetahui fungsi apa saja yang dapat dilakukan oleh aplikasi. Kebutuhan fungsional dari Pengembangan *REST API* pada Aplikasi SAKU ini adalah

1. Aplikasi ini memiliki tiga *level user*, yaitu pengguna, *admin*, dan *super-admin*.
2. Aplikasi ini menyediakan beberapa fitur bagi pengguna yaitu mengelola data transaksi, melihat laporan, mengunduh laporan dan mengelola data perusahaan.
3. Aplikasi ini menyediakan fitur bagi *admin* yaitu manajemen *user*.
4. Aplikasi ini menyediakan fitur bagi *super-admin* yaitu manajemen *admin*.

Kebutuhan fungsional aplikasi digambarkan dalam *use case diagram* pada Gambar 3.



Gambar 3. *Use Case Diagram* Aplikasi SAKU.

### 3.3.2 Kebutuhan *Non-Fungsional*

Kebutuhan *non fungsional* merupakan kebutuhan yang berada di luar fungsi-fungsi yang dapat dilakukan oleh aplikasi yaitu kebutuhan yang berhubungan dengan lingkungan aplikasi, keandalan aplikasi, keamanan aplikasi, dan kinerja aplikasi.

#### 3.3.2.1 Kebutuhan Pengguna

Berikut adalah Kebutuhan pengguna dalam penelitian ini

1. Untuk mengakses beberapa *endpoint* di API diperlukan token pada *header* untuk memvalidasi *user* sudah *login*.
2. Hanya akun yang sudah melakukan verifikasi *email* yang dapat *login* ke dalam sistem.

#### 3.3.2.2 Kebutuhan Kinerja

Berikut ini adalah kebutuhan kinerja dalam penelitian ini

1. Aplikasi memiliki keamanan yang baik dilengkapi dengan *password login*.
2. Penyimpanan dan pengolahan data cepat, tepat dan efisien.

#### 3.3.2.3 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah laptop dengan spesifikasi sebagai berikut :

1. *Processor* : AMD Ryzen 5 4500U.
2. *Memory/RAM* : 16GB DDR4.
3. *Hardisk* : 512GB SSD.
4. *System Type* : 64-bit operating system, x64-based *processor*.

### 3.3.2.4 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

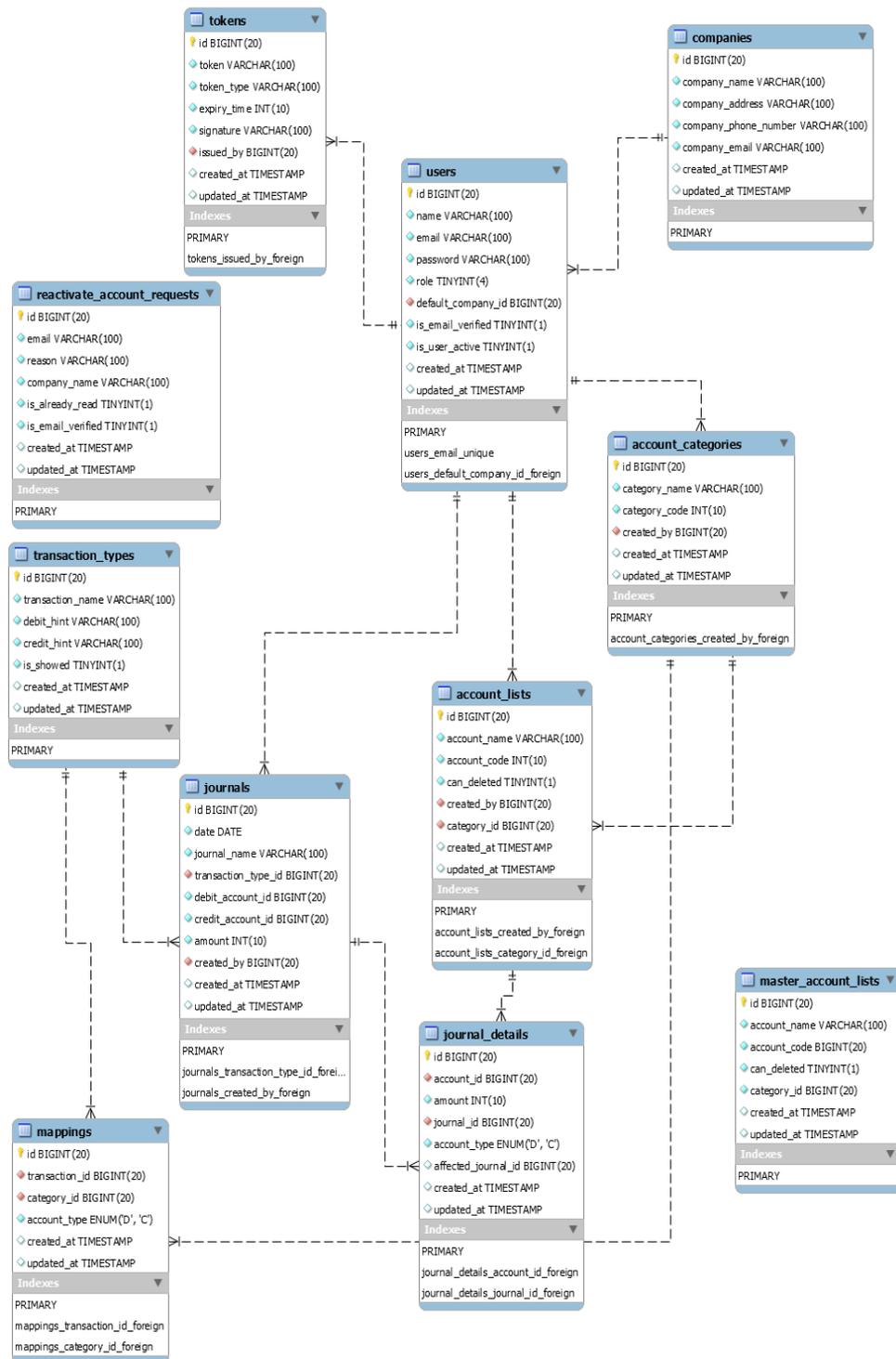
1. Sistem Operasi *Microsoft Windows* 10 64 Bit.
2. XAMPP versi 3.2.4.
3. *Microsoft Visual Studio Code* versi 1.5.8 sebagai IDE.
4. *Git* sebagai *version control*.
5. *Framework Lumen* versi 8.2.3 sebagai *framework* dalam pembuatan *REST API*.
6. *Postman* versi 8.7.0 sebagai *REST Client* untuk uji coba *REST API*.

## 3.4 Perancangan Database

Tahap perancangan *database* akan dilakukan dengan menggunakan bantuan *entity relationship diagram* (ERD) dan dilakukan berdasarkan kebutuhan pengguna dan analisis kebutuhan sistem. Analisis kebutuhan sistem terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional.

Setiap kebutuhan pengguna yang telah didapatkan dalam tahap analisis kebutuhan sistem, kemudian akan diterjemahkan menjadi entitas atau class untuk mewakili himpunan objek yang memiliki atribut dan operasi yang sama, kemudian setiap entitas akan saling terhubung dengan relasi yang menunjukkan hubungan antar setiap entitas yang ada di dalam *database* .

*Database* akan dirancang untuk tiga level pengguna yakni *super-admin*, *admin*, dan *user* dengan memperhatikan cakupan fungsi manajemen yang dapat dijalankan oleh setiap tingkatan pengguna. Hasil pembentukan ERD dengan menggunakan bantuan *entity relationship diagram* (ERD) berdasarkan analisis kebutuhan sistem disajikan pada Gambar 4.



Gambar 4. ERD Aplikasi Solusi Akuntansi UMKM (SAKU).

Berdasarkan Gambar 4, dapat dilihat terdapat sebelas entity yang akan merupakan hasil perancangan *database* yang akan digunakan dalam aplikasi SAKU.

### 3.5 Skenario Pengujian

Dalam penelitian ini, akan digunakan pengujian sistem dengan menggunakan metode *black-box testing*. Jenis *black-box testing* yang digunakan pada penelitian ini adalah *Equivalence Partitioning*. *Equivalence Partitioning* merupakan metode pengujian pada *black-box* dengan menggunakan skenario uji, hasil yang diharapkan dan hasil pengujian untuk melihat apakah sistem berjalan sesuai atau tidak. pengujian API pada *super-admin*, *admin*, *user* dan keamanan menggunakan URI (*Uniform Resource Identifier*) yang sudah tersedia. Contoh skenarionya dapat dilihat pada Tabel 3 (Pranata dkk., 2018).

Tabel 3 Contoh Skenario Pengujian *Equivalence Partitioning* Pada API

| <b>Level User</b>                  | <b>Kasus Uji</b>                           | <b>Daftar Pengujian</b> | <b>Hasil yang Diharapkan</b> | <b>Status</b>       |
|------------------------------------|--|-------------------------|------------------------------|---------------------|
| <i>Super-admin, Admin dan User</i> | Request Method (GET, POST, PUT dan DELETE) | Validasi Path (URI)     | Sukses menampilkan           | Berhasil atau Gagal |

Sebagaimana disajikan dalam Tabel 3, skenario pengujian akan dilakukan untuk setiap level pengguna, yakni *Super-admin*, *Admin*, dan *User*. Kemudian, hasil yang didapatkan akan menggambarkan apakah sistem berhasil untuk menjalankan setiap skenario pengujian yang dilakukan untuk setiap level pengguna.

## V. SIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan :

- 5.1.1 REST API yang dibangun untuk aplikasi SAKU dapat digunakan pada *platform* Website dan Android.
- 5.1.2 REST API pada Aplikasi SAKU ini Memiliki tiga level *user*, yaitu *super-admin*, *admin*, dan pengguna UMKM. *Super-admin* dapat melihat seluruh daftar *admin*, menambah, mengubah dan menghapus data *admin*.

Level pengguna *admin* dapat melihat seluruh *user* aktif dan pasif, mengubah, menghapus data *user*, mengaktifkan ulang *user* yang sebelumnya tidak aktif, dan menampilkan daftar permintaan pengaktifan kembali, mengaktifkan kembali akun berdasarkan permintaan, membuat tandai pesan yang telah dibaca, menampilkan daftar akun master data, menambah master data perkiraan, memperbarui master data perkiraan, menghapus master data perkiraan.

Level pengguna *user* atau pengguna UMKM dapat menambahkan, menghapus, dan mengedit transaksi, melihat laporan jurnal, buku besar, neraca saldo, laba rugi, perubahan modal, neraca, dan hutang piutang, mengelola data perusahaan dan kode rekening, menghapus akun, dan mengubah kata sandi.

## 5.2 Saran

Berdasarkan kesimpulan yang didapat, menunjukkan bahwa dapat dibuat saran-saran yang akan membantu untuk pengembangan lebih lanjut dari REST API pada Aplikasi SAKU, yaitu :

- 5.2.1 Mengembangkan manajemen sesi untuk setiap level pengguna yang meliputi *super-admin*, *admin*, dan *user*, untuk meningkatkan keamanan dan pemantauan aktivitas yang menggunakan API.
- 5.2.2 Mengembangkan fitur *error logging*, untuk melakukan pendataan dan membantu visualisasi apabila terjadi kesalahan sistem.
- 5.2.3 Mengembangkan fitur limit rates untuk API aplikasi SAKU, yang berfungsi untuk mengatasi *spam* yang dilakukan aktor diluar sistem dan mengurangi beban server.

# **DAFTAR PUSTAKA**

## DAFTAR PUSTAKA

- Abdurahman, H., dan Riswaya, A. R. (2014). Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha BHakti. *Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha BHakti*.
- Firman, A., Wowor, H. F., dan Najoran, X. (2016). Sistem Informasi Perpustakaan Online Berbasis Web. *E-Journal Teknik Elektro Dan Komputer*, 5(2), 29–36.
- Fridayanthie, E. W., dan Mahdiati, T. (2016). Rancang Bangun Sistem Informasi Permintaan Atk Berbasis Intranet (Studi Kasus: Kejaksaan Negeri Rangkas Bitung). *Resma*, 3(2), 13–22.
- Halili, F., and Ramadani, E. (2018). Web Services: A Comparison of Soap and Rest Services. *Modern Applied Science*, 12(3), 175. <https://doi.org/10.5539/mas.v12n3p175>
- Undang-Undang Republik Indonesia Nomor 20 Tentang Usaha Mikro, Kecil, dan Menengah*, (2008) (testimony of Republik Indonesia).
- Latif, A. (2015). Implementasi Kriptografi Menggunakan Metode Advanced Encryption Standar (Aes) Untuk Pengamanan Data Teks Implementasi Kriptografi Menggunakan Metode Advanced Encryption Standar (Aes) Untuk Pengamanan Data Teks. *Jurnal Ilmiah Mustek Anim Ha - ISSN 2089-6697*, 13(3), 1576–1580.
- Martani, D., Siregar, S. V., Wardhani, R., Farahmita, A., dan Tanujaya, E. (2016). Akuntansi Keuangan Menengah Berbasis PSAK. In *Salemba Empat*.
- Mulyanto, A. R. (2018). *Rekayasa Perangkat Lunak Jilid 2* (Vol. 9, Issue 4).
- Perdana, M. A. K. (2018). Pengembangan REST API Layanan Penyimpanan menggunakan Metode Rapid Application Development (Studi kasus PT. XYZ). *InfoTekJar (Jurnal Nasional Informatika Dan Teknologi Jaringan)*, 3(1), 100–104.
- Pranata, B. A., Hijriani, A., dan Junaidi, A. (2018). Perancangan Application Programming Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem

- Informasi Administrasi Pasien Klinik Perawatan Kulit. *Jurnal Komputasi*, 6(1), 33–42. <https://doi.org/10.23960/komputasi.v6i1.1554>
- Pressman, R. S., and Maxim, B. R. (2020). *Software Engineering: A PRACTITIONER'S APPROACH 9TH Edition*. *Kybernetes*, 29(9/10), 8–9. <https://doi.org/10.1108/k.2000.06729iag.002>
- Putra, D. W. T., dan Andriani, R. (2019). Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. *Jurnal TeknolIf*, 7(1), 32. <https://doi.org/10.21063/jtif.2019.v7.1.32-39>
- Qibtiyah, U. M., dan Rahayu, S. (2017). Implementasi JSON Web Service Pada Aplikasi Digital Library Politeknik Sukabumi. *JTERA - Jurnal Teknologi Rekayasa*, 3(1), 9–16.
- Rahmatulloh, A., Sulastri, H., dan Nugroho, R. (2018). Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, 7(2). <https://doi.org/10.22146/jnteti.v7i2.417>
- Surahman, F., Al Ikhsan, H., dan Kusumah, F. (2018). Rancang Bangun Web Service untuk Transaksi Data pada Aplikasi Sahabat Jasa dengan Metode REST. *Seminar Nasional Teknologi Informasi*, 1, 256–264.
- Wahyudi, R., Utami, E., dan Arief, M. R. (2016). Sistem Pakar E-tourism pada Dinas Pariwisata DIY Menggunakan Metode Forward Chaining. *Jurnal DASIS*, 17.