

STUDI KELAYAKAN SISTEM KOMPUTER UNTUK PEMBANGUNAN LAYANAN SISTEM OTENTIKASI DI BADAN PENGELOLAAN PENDAPATAN, KEUANGAN DAN ASET DAERAH KABUPATEN BREBES

20 Februari 2019

Priyanto Tamami, S.Kom.

1 RUANG LINGKUP PEKERJAAN

Ruang lingkup dari pekerjaan studi kelayakan sistem komputer ini hanya sebatas menilai apakah sistem komputer yang sudah ada layak atau tidak untuk digunakan sebagai basis layanan sistem otentikasi di Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah Kabupaten Brebes.

Aplikasi atau sistem otentikasi yang akan dibangun diatas sistem komputer ini nantinya akan dibangun berbasis *web*, sehingga dari sisi peladen (*server*) nantinya, sistem komputer yang terpasang harus mampu baik secara perangkat keras (*hardware*) maupun perangkat lunak (*software*), memberikan layanan bagi setiap komputer klien yang terpasang baik di kantor maupun di masyarakat umum. Sedangkan dari sisi komputer klien, lingkupnya hanya sebatas komputer klien yang terdapat di bidang Pendataan dan Penetapan pada Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah Kabupaten Brebes, apakah sistem komputer yang terpasang, baik secara perangkat keras (*hardware*) dan perangkat lunaknya (*software*) mampu untuk melakukan akses ke peladen (*server*) dari layanan (*service*) *OAuth Server*.

Karena aplikasi yang dibangun di atas sistem komputer di sisi peladen (*server*) ini akan berbasis *container* docker, tentunya perangkat lunak docker wajib

terpasang, sedangkan di sisi komputer klien, yang dibutuhkan hanyalah sebuah peramban (*browser*) untuk melakukan akses ke layanan di sisi peladen (*server*).

Hal-hal yang mendukung studi kelayakan sistem komputer yang dibutuhkan oleh sistem otentikasi di Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah ini yaitu dengan menggunakan metode analisis kelayakan TELOS (*Technical, Economic, Legal, Operational, Schedule*).

Detail dari masing-masing faktor pada metode analisis kelayakan TELOS ini adalah seperti berikut :

1. Faktor Kelayakan Teknis (*Technical*)

Kelayakan teknis menyoroti kebutuhan sistem yang telah disusun dari aspek teknologi yang digunakan, jika teknologi yang dikehendaki untuk pengembangan sistem merupakan teknologi yang mudah didapat, murah, dan tingkat pemakaiannya mudah, maka secara teknis usulan kebutuhan sistem bisa dinyatakan layak.

2. Faktor Kelayakan Ekonomi (*Economic*)

Aspek yang paling dominan dari aspek kelayakan yang lain adalah kelayakan ekonomi. Tidak dapat disangkal lagi, motivasi pengembangan sistem informasi pada perusahaan atau organisasi adalah motif keuntungan. Dengan demikian aspek untung rugi jadi pertimbangan utama dalam pengembangan sistem, termasuk di dalamnya adalah kelayakan ekonomi atas sistem komputer yang akan digunakan. Kelayakan ekonomi berhubungan dengan *return investmen* atau berapa lama biaya investasi dapat kembali, namun pada instansi pemerintah, yang secara umum bertugas melayani masyarakat, maka hubungannya bukan lagi untung rugi, tetapi lebih kepada imbas administrasi data yang tercatat dengan baik dan sedikit banyak mungkin berpengaruh

pada peningkatan kepatuhan wajib pajak dalam memenuhi kewajiban pajaknya.

3. Faktor Kelayakan Hukum (*Legal*)

Menguraikan secara hukum apakah sistem komputer yang dibutuhkan tidak menyimpang dari hukum yang berlaku (tidak melanggar hukum jika diterapkan di objek penelitian). Misal: bagaimana kelayakan perangkat lunak yang digunakan, bagaimana kelayakan hukum informasi yang dihasilkan oleh program aplikasi yang dibuat. Apakah melanggar hukum atau tidak.

4. Faktor Kelayakan Operasional (*Operational*)

Penilaian terhadap kelayakan operasional digunakan untuk mengukur apakah sistem yang akan dikembangkan nantinya dapat dioperasikan dengan baik atau tidak untuk melayani kebutuhan otentikasi dan otorisasi.

5. Faktor Kelayakan Jadwal (*Schedule*)

Penilaian kelayakan jadwal ini digunakan untuk menentukan bahwa pengembangan sistem akan dapat dilakukan dalam batas waktu yang telah ditetapkan.

Adapun penilaian untuk masing-masing faktor kelayakan TELOS ini adalah sebagai berikut :

1. Menilai kelayakan Teknik (*Technical*)

Kelayakan teknik menyoroti kebutuhan sistem yang telah disusun dari teknologi yang akan digunakan, untuk penerapan sistem otentikasi di Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah. Sistem ini merupakan sistem berbasis layanan otentikasi yang digunakan oleh aplikasi yang akan

dibangun yang datanya menggunakan data-data Pajak Daerah dan akses datanya terbatas untuk tiap jabatan pengguna, sehingga di masa mendatang, aplikasi yang dibangun tidak memerlukan bagian otentikasi, proses otentikasi cukup diserahkan ke sistem OAuth yang akan dibangun ini.

Sistem ini nantinya akan menerima permintaan sebuah token dari aplikasi klien, yang apabila belum pernah masuk sama sekali, sistem akan menampilkan halaman masuk (*login*) untuk memastikan bahwa pengguna (*end-user*) sudah terdaftar dan memiliki hak untuk melakukan akses data. Apabila proses *login* telah berhasil, maka aplikasi klien akan mendapatkan sebuah token yang dapat digunakan untuk melakukan akses data ke *resource server*.

Token yang diberikan oleh *OAuth server* berbentuk JWT (*JSON Web Token*) yang akan digunakan oleh aplikasi klien melakukan akses ke *resource server*. Token dalam bentuk JWT ini sebetulnya sudah memiliki informasi yang cukup bagi aplikasi klien melakukan pengaturan akses bagi pengguna yang sedang melakukannya, tentunya aplikasi klien perlu melakukan *decode* terlebih dahulu pada JWT yang diterima dari *OAuth Server*.

Token dalam bentuk JWT ini wajib selalu dikirimkan oleh aplikasi klien pada saat melakukan akses ke *resource server*, hal ini untuk memastikan bahwa data yang berada di *resource server* diakses oleh pengguna yang memiliki hak.

(a) Kebutuhan Perangkat Keras, Perangkat Lunak, dan Perangkat Jaringan

i. Perangkat Keras

Kebutuhan akan perangkat keras, sebetulnya tidak diperlukan pengadaan alat baru, melainkan dengan menggunakan perangkat yang sudah ada, sistem informasi ini akan dapat berjalan sebagaimana

mestinya. Adapun perangkat keras yang digunakan adalah seperti berikut :

No	Perangkat keras	Keterangan
<i>Server Aplikasi</i>		
1	Prosesor	Intel Xeon 2,4G 8 Core
2	Memori	8 GB
3	Harddisk	
4	Jaringan	4 ethernet slot
<i>Server Basis Data</i>		
1	Prosesor	Intel Xeon 2,4G 8 Core
2	Memori	32 GB
3	Harddisk	
4	Jaringan	4 ethernet slot

ii. Perangkat Lunak

Kebutuhan akan perangkat lunak baru tentunya ada, tetapi tidak memerlukan biaya tambahan, karena beberapa perangkat lunak ada yang sudah terpasang, dan perangkat lunak yang akan di*install* tidak memerlukan biaya tambahan.

No	Perangkat Lunak	Kegunaan
1	Windows Server 2008	Sistem Operasi
2	Linux Ubuntu	Sistem Operasi
3	IntelliJ IDEA	IDE
4	Visual Studio Code	Editor
5	Docker	Container
6	Chrome, Firefox	Web Browser

iii. Perangkat Jaringan

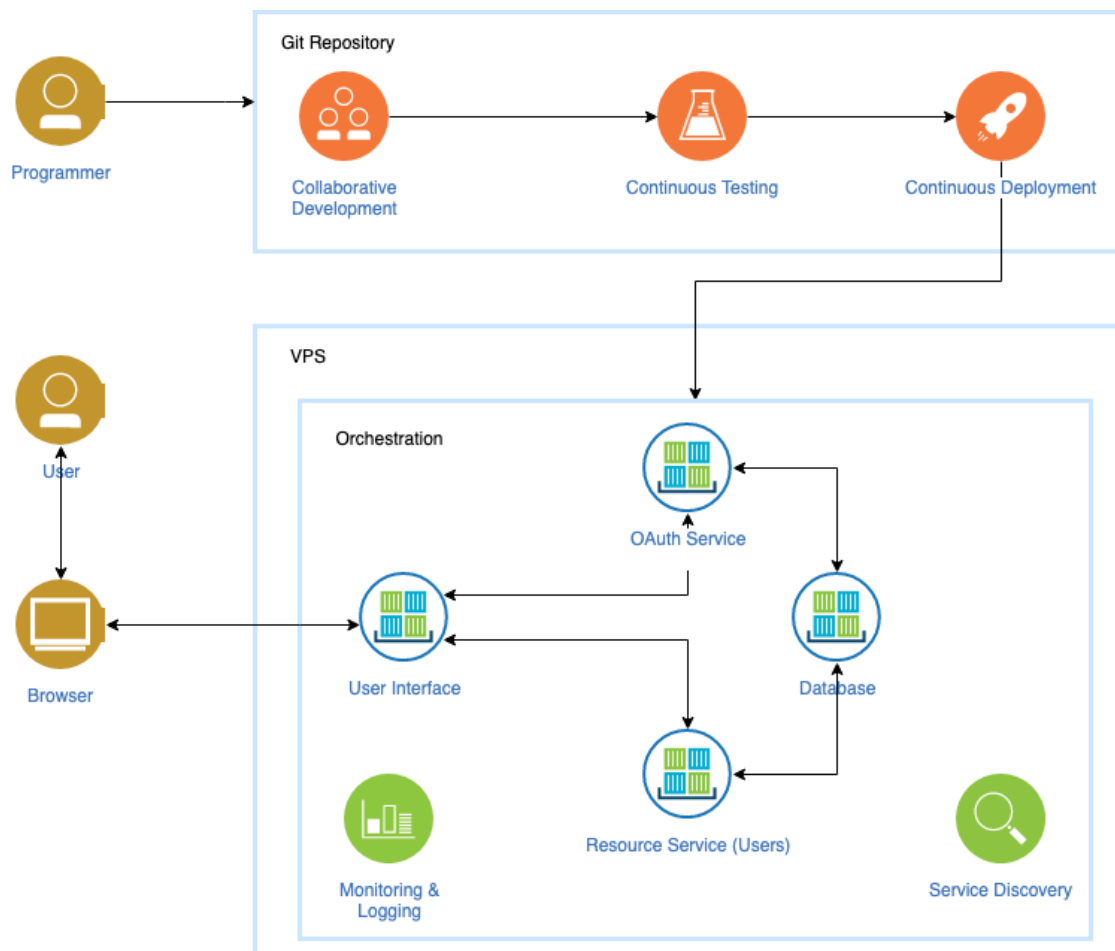
Kebutuhan akan perangkat jaringan pun tidak memerlukan pengadaan kembali, karena kondisi jaringan sudah terhubung dengan baik dengan peralatan seperti berikut :

No	Perangkat	Kegunaan
1	Switch	Penghubung antar kabel jaringan
2	Kabel UTP	Media penghubung antar perangkat
3	Konektor RJ-45	Media penghubung antara kabel dengan <i>socket</i> perangkat
4	Modem	Sebagai perangkat penghubung jaringan lokal dan internet
5	Router	Penghubung antar 2 atau lebih jaringan (berdasarkan alamat jaringan)

(b) Arsitektur Jaringan Komputer

Arsitektur jaringan sistem informasi ini adalah seperti pada gambar

1. Bila ada seorang pengguna yang melakukan akses dengan *browser*, maka akan ditampilkan halaman utama dari sistem informasi ini.



Gambar 1: Desain Arsitektur Sistem

Nantinya, saat pengguna akan melakukan akses terhadap beberapa aplikasi yang terproteksi, pengguna akan diarahkan ke halaman *login* milik *OAuth Service* untuk mengisi *username* dan *password*.

Setelah proses otentikasi berhasil dilakukan, selanjutnya halaman pada peramban (*browser*) akan dikembalikan ke halaman aplikasi yang akan diakses, dimana proses yang berjalan dibelakang layar adalah *OAuth Service* akan memberikan sebuah *token* ke aplikasi klien untuk disimpan dan digunakan setiap akan melakukan proses permintaan data (*request*)

ke *Resource Service*.

Selanjutnya, saat pengguna akan melakukan akses data melalui halaman aplikasi *web* klien, aplikasi ini akan melakukan permintaan (*request*) data ke *resource service* dengan membawa data dan *token*, yang kemudian *resource service* akan melakukan verifikasi hak akses melalui *token* tersebut, apabila hak akses tidak sesuai, maka akan mengembalikan pesan kesalahan bahwa permintaan atau *request* tersebut tidak dapat dipenuhi, namun bila hak aksesnya sesuai, maka data akan dikirimkan ke aplikasi klien untuk ditampilkan ke pengguna.

Adapun kondisi sistem informasi yang digunakan saat ini adalah seperti berikut :

(a) Sistem Aplikasi

Aplikasi adalah sebuah *tools* untuk membantu dan mempermudah suatu aktifitas administrasi maupun yang lainnya. Tanpa adanya aplikasi, tentu suatu aktifitas tidak bisa diselesaikan dengan cepat, segala sesuatunya dilakukan dengan pencatatan manual. Aplikasi atau sistem informasi yang digunakan di Bidang Pendataan dan Penetapan pada Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah Kabupaten Brebes saat ini adalah :

No	Aplikasi	Keterangan
1	SISMIOP	Melakukan manajemen objek pajak PBB-P2
2	Simda Pendapatan	Melakukan pengelolaan pajak daerah
3	Microsoft Excel	Digunakan untuk merekam dan mengolah data peralihan hak

Dengan adanya sistem informasi atau aplikasi SISMIOP, sudah membantu pengelolaan pajak daerah untuk jenis Pajak Bumi dan Bangunan

sektor Perdesaan dan Perkotaan dari mulai manajemen pelayanan, pendataan, penetapan, penagihan, hingga ke prosedur pembayaran.

Sedangkan aplikasi Simda Pendapatan, sudah dapat membantu melakukan pengelolaan pajak daerah untuk jenis pajak selain Pajak Bumi dan Bangunan sektor Perdesaan dan Perkotaan, namun belum sepenuhnya dapat melakukan pengelolaan yang terjadi pada jenis Bea Perolehan Hak Atas Tanah dan Bangunan.

Dengan semakin bertambahnya kebutuhan organisasi akan aplikasi atau sistem informasi kedepan, maka akan semakin bertambah pula otentikasi dan otorisasi untuk tiap penggunaanya, untuk mengantisipasi kondisi seperti ini, maka kebutuhan untuk membangun sebuah sistem OAuth diperlukan sehingga nantinya tiap pengguna tidak perlu melakukan *login* berulang-ulang untuk melakukan akses ke beberapa aplikasi yang terproteksi.

(b) Sistem Basis Data

Basis data merupakan suatu tempat penyimpanan data-data yang biasanya dari sebuah aplikasi, adapun basis data yang sudah ada pada Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah adalah seperti berikut :

No	Basis data	Keterangan
1	Oracle DB	Digunakan untuk menyimpan data dari aplikasi SISMIOP
2	SQL Server	Digunakan untuk menyimpan data dari aplikasi Simda Pendapatan

Basis data yang digunakan masih belum terintegrasi dengan baik, karena penerimaan pembayaran pada SISMIOP belum dapat secara otomatis tercatat pada Simda Pendapatan.

Sebetulnya sumber data yang digunakan pada pengelolaan Pajak Bumi dan Bangunan sektor Perdesaan dan Perkotaan ada pada 2 (dua) tempat, yang pertama berada di Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah Kabupaten Brebes, yang kedua berada di BPD Jateng.

Kondisi data yang berada di Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah berfungsi sebagai pendataan dan penetapan, namun realisasi pembayaran menggunakan data dari BPD Jawa Tengah. Selisih nilai antara 2 (dua) sumber data ini pasti terjadi, maka seharusnya dibutuhkan personil untuk melakukan pencocokan / rekonsiliasi data antara 2 (dua) sumber data tersebut agar data yang ditampilkan pada sistem informasi yang akan dibangun menjadi lebih lengkap.

Untuk membangun sistem OAuth nantinya membutuhkan sistem basis data yang tentunya terpisah dari sistem yang telah ada, penggunaan sistem basis datanya pun tidak memerlukan biaya tambahan karena menggunakan sistem basis data yang gratis dan sudah teruji kehandalannya. Basis data ini nantinya akan merekam daftar pengguna dan kata kuncinya (*password*) termasuk statusnya apakah sebagai pengguna yang statusnya aktif, atau pengguna yang statusnya di non-aktifkan, yang artinya tidak dapat melakukan akses masuk ke data-data yang terproteksi.

Pada sistem basis data yang digunakan OAuth ini pun akan merekam data aplikasi klien yang terdaftar yang diijinkan melakukan akses ke *resource server* yang terproteksi.

Dilihat dari sisi teknis sistem basis data yang akan digunakan cukup handal, terbukti dengan banyaknya perusahaan besar menggunakan basis data ini dan tersedia secara gratis, maka syarat kebutuhannya sangat

mencukupi untuk membangun sistem OAuth.

(c) Infrastruktur

Infrastruktur merupakan sarana yang sangat dibutuhkan dalam sebuah aktifitas apapun termasuk aktifitas pengelolaan Pajak Daerah. Adapun infrastruktur yang ada pada Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah Bidang Pendataan dan Penetapan yaitu :

No	Infrastruktur	Keterangan
1	Komputer	Memiliki 13 komputer dan 4 <i>server</i>
2	Printer	Memiliki 4 <i>printer multi-purpose</i> , 4 <i>printer dot-matrix</i> , 3 <i>printer line-matrix</i>
3	Jaringan Internet	Memiliki 2 (dua) sumber internet, dengan 1 (satu) internet <i>shared bandwidth</i> dan 1 (satu) internet <i>dedicated</i>

Dari kondisi infrastruktur di atas, sebetulnya Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah secara infrastruktur, syarat kebutuhannya sangat mencukupi untuk membangun sistem OAuth yang dapat digunakan sebagai pendukung aplikasi yang akan dibangun kedepannya.

Melihat kondisi teknis demikian, dimana sistem komputer yang telah ada dianggap mampu melakukan tugasnya untuk penerapan aplikasi atau sistem otentikasi, sehingga nilai kelayakan teknik yang dapat diberikan adalah 9.0.

2. Menilai kelayakan Ekonomi (*Economic*)

Pembangunan sistem baru tentunya membutuhkan investasi ataupun dana yang tidak sedikit, untuk mendapatkan manfaat di masa yang akan datang, sumber daya dan sumber dana diperlukan dalam pembangunan sistem baru sebagai bentuk investasi.

Untuk menganalisis kelayakan ekonomi digunakan kalkulasi analisis biaya dan manfaat (*cost benefit analysis*). Adapun tujuan dari analisis biaya dan manfaat adalah untuk memberikan gambaran kepada internal Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah, apakah manfaat yang diperoleh dari sistem baru "lebih besar" dibandingkan dengan biaya yang dikeluarkan.

Pada analisis biaya dan manfaat ada beberapa metode kuantitatif yang digunakan untuk menemukan standar kelayakan proyek.

Analisis Biaya dan Manfaat

Untuk melakukan analisa biaya dan manfaat diperlukan dua komponen, yaitu komponen biaya dan komponen manfaat.

(a) Komponen Biaya

Biaya yang berhubungan dengan pembuatan sistem ini dapat diklasifikasikan ke dalam 3 (tiga) kategori utama, yaitu :

- i. Biaya pengadaan, yaitu biaya pembelian perangkat keras, yang digunakan pada awal pembuatan sistem, sebelum sistem dioperasikan.
- ii. Biaya Pengembangan, yaitu biaya pembuatan perangkat lunak sistem yang meliputi biaya konsultasi, biaya tahap analisa sistem, biaya tahap desain sistem dan biaya tahap penerapan sistem.
- iii. Biaya operasi dan biaya perawatan, yaitu biaya yang dikeluarkan untuk menjalankan sistem, yaitu biaya *overhead*, biaya perawatan terhadap perangkat keras dan perangkat lunak.

(b) Komponen Manfaat.

Manfaat yang didapat dari sistem informasi diklasifikasi sebagai berikut :

- i. Keuntungan berwujud adalah keuntungan yang berupa penghematan atau peningkatan di dalam pengelolaan pajak daerah yang dapat diukur dalam bentuk satuan nilai uang. Keuntungan berwujud antara lain :
 - A. Pengurangan biaya cetak
 - B. Pengurangan biaya operasi
 - C. Pengurangan biaya perlengkapan
- ii. Keuntungan tak berwujud, adalah keuntungan yang sulit atau tidak mungkin diukur dalam bentuk satuan uang. Keuntungan tersebut antara lain :
 - A. Peningkatan efisiensi waktu pengerjaan
 - B. Peningkatan efektifitas pegawai
 - C. Peningkatan pelayanan pajak daerah

Adapun metode untuk melakukan analisis biaya dan manfaat adalah :

- i. Metode Periode Pengembalian

Metode ini adalah uji kuantitatif yang digunakan untuk menghitung jangka waktu yang diperlukan untuk membayar kembali biaya investasi dalam pembuatan aplikasi yang telah dikeluarkan. Penilaian kelayakan untuk pengembalian

$$periode = \frac{investasi}{proceed} \times tahun$$

- A. Layak jika waktu pengembalian lebih kecil dari umur investasi
- B. Tidak layak jika waktu pengembalian lebih besar dari umur investasi.

Perhitungan Periode Pengembalian adalah seperti berikut :

Nilai Investasi = Rp0,- Proses Th 1 = Rp100.364.100.000,-

Nilai investasi memang bernilai 0 (nol) rupiah karena seluruh sarana dan prasarana telah tersedia, hanya tinggal membangun sebuah sistem informasi untuk digunakan dan dijalankan. Sedangkan nilai Rp100.364.100.000,- (Seratus milyar tiga ratus enam puluh empat juta seratus ribu) didapat dari nilai target penerimaan untuk seluruh jenis Pajak Daerah di Kabupaten Brebes.

$$PP = \frac{0}{100.364.100.000}$$

$$PP = 0 \text{ tahun}$$

Artinya, karena nilai investasi yang dikeluarkan nihil sama sekali, atau dengan kata lain tidak memerlukan nilai investasi, namun sistem informasi secara tidak langsung memberikan andil terhadap realisasi sebesar kurang lebih 100 Milyar Rupiah kepada Kas Daerah. Artinya sistem ini sangat layak untuk dikembangkan karena pengembalian tidak membutuhkan waktu untuk mencapai titik impas.

ii. Metode Pengembalian Investasi

Metode pengembalian investasi digunakan untuk mengukur persentase manfaat yang dihasilkan proyek dibandingkan dengan biaya yang dikeluarkan.

Return On Investmen (ROI) dari suatu proyek dapat dihitung dengan rumus penilaian kelayakan untuk ROI :

A. Layak jika $ROI > 0$

B. Tidak layak jika $ROI < 0$

$$ROI = \frac{TotalManfaat - TotalBiaya}{TotalBiaya} \times 100\%$$

Karena nilai biaya tidak akan pernah muncul, maka bisa disebut bahwa Total Biaya = 0 (nol), Total Manfaat adalah penerimaan realisasi untuk seluruh jenis pajak daerah dalam satu tahun pajak, yaitu senilai 100.364.100.000 rupiah. Maka hasil yang didapatkan adalah seperti berikut :

$$ROI = \frac{100.364.100.000 - 0}{0} \times 100\%$$

$$ROI = \infty$$

Ternyata karena kondisi pembagiya adalah 0 (nol), maka hasil yang didapatkan menghasilkan Tak Terhingga. Ini dikarenakan bilangan pembagi adalah 0 (nol) sehingga menghasilkan nilai demikian, artinya proyek apapun itu bila tidak memerlukan biaya sama sekali, namun bisa mendapatkan manfaat maksimal tentunya ini proyek dengan ROI tak terhingga.

Jadi, karena nilai ∞ itu lebih dari 0 (nol) maka proyek ini layak untuk dikerjakan.

iii. Metode Nilai Sekarang Bersih

Metode nilai sekarang bersih merupakan metode yang memperhatikan nilai waktu dari uang. Suku bunga diskonto mempengaruhi arus dari uangnya (*proceed*). Metode Nilai Sekarang Bersih atau lebih dikenal dengan *Net Present Value* (NPV) dapat dihitung dari selisih nilai proyek pada awal tahun dikurangi dengan *proceed* tiap tahun yang dinilai uangkan ketahun awal dengan tingkat bunga diskonto.

Kriteria NPV ini adalah seperti berikut :

- $NPV > 0$: *Feasible*

- $NPV = 0$: *Indifferent*
- $NPV < 0$: *Unfeasible*

Rumus untuk menghitung NPV ini adalah seperti berikut :

$$NPV = -NilaiProyek + \frac{proceed1}{(1+i)^1} + \frac{proceed2}{(1+i)^2} + \frac{proceedn}{(1+i)^n}$$

dimana :

- i adalah tingkat bunga diskonto diperhitungkan
- n adalah umur proyek investasi
- $proceed$ adalah selisih biaya dan manfaat

Sehingga perhitungannya dapat kita cari seperti berikut :

$$NPV = -0 + \frac{100.364.100.000}{(1 + 6,0\%)^1}$$

$$NPV = -0 + \frac{100.364.100.000}{1,06}$$

$$NPV = 94.683.113.207,55$$

Pada perhitungan di atas, nilai waktu dari bunga uang yang ditanamkan adalah 6,0% berdasarkan suku bunga dari www.bi.go.id pada tanggal 17 Januari 2019. Karena nilai $NPV > 0$ berarti investasi dapat diterima dan layak untuk dikembangkan.

Dengan melihat ketiga metode untuk kelayakan ekonomi maka dapat dikatakan bahwa sistem komputer yang akan sudah ada layak dan dapat digunakan sebagaimana mestinya. Nilai yang diberikan untuk kelayakan ekonomi ini adalah 9,0.

3. Menilai kelayakan Hukum (*Legal*)

Kelayakan hukum adalah kelayakan yang berkaitan dengan legalitas atau kekuatan hukum. Berarti bahwa sistem informasi yang akan dibangun tidak boleh melanggar hukum yang berlaku, baik hukum yang ditetapkan oleh pemerintah maupun hukum yang ditetapkan berdasarkan peraturan-peraturan organisasi.

Proyek pembangunan sistem informasi yang akan dikembangkan secara hukum dinilai layak karena perangkat lunak (*software*) yang digunakan resmi sesuai dengan perijinan yang ada, karena sebagian besar menggunakan perangkat lunak yang bersifat *free* atau gratis.

Adapun rincian perangkat lunak secara hukum adalah sebagai berikut :

No	<i>Software</i>	Lisensi
1	Windows Server 2008 R2	Lisensi tersedia
2	Linux Ubuntu	Gratis
3	IntelliJ IDEA Community Edition	Gratis, Apache 2.0
4	Visual Studio Code	Gratis, MIT
5	Docker	Gratis, Apache 2.0
6	Git	Gratis, LGPLv2.1

Dengan informasi seperti di atas bahwa kelayakan hukum dari sistem komputer yang telah ada tentunya layak untuk digunakan dan diberikan nilai 9,0.

4. Menilai kelayakan Operasional (*Operational*)

Kelayakan operasional dinilai dengan menggunakan kerangka kerja PIECES yang merupakan singkatan dari *Performance*, *Information*, *Economy*, *Control*, *Efficiency*, dan *Services*. Kerangka kerja PIECES ini dikembangkan

oleh James Wetherbe yang bertujuan mengukur apakah sistem yang akan dikembangkan dapat dioperasikan dengan baik atau tidak di dalam organisasi.

Penjelasan untuk masing-masing poin dalam kerangka kerja PIECES ini adalah seperti berikut ini :

- *Performance* atau kinerja adalah untuk mengetahui apakah sistem menyediakan *throughput* dan *response time* yang cukup. Dari sisi *performance* atau kinerja, sistem komputer yang telah ada masih layak digunakan untuk dipasangkan suatu sistem atau aplikasi otentikasi karena penggunaan memori yang baru digunakan sebesar 80% dari kapasitas total sebesar 8 GB, sedangkan dari sistem jaringan pun yang mendukung 1 GB/s hanya digunakan untuk keperluan akses data yang sangat kecil.
- *Information* atau informasi adalah untuk mengetahui apakah sistem komputer menyediakan informasi yang berkualitas bagi pengguna. Dari sistem komputer yang telah terpasang sebetulnya sudah dapat menyajikan informasi yang cukup bahkan pada saat kondisi peladen (*server*) mati pun dapat kita monitor dari aplikasi iLO milik HP, informasinya dapat kita peroleh baik secara *on-site* ataupun *remote*.
- *Economy* atau ekonomi adalah untuk mengetahui apakah sistem menawarkan tingkat dan kapasitas pelayanan yang memadai untuk mengurangi biaya dan meningkatkan keuntungan. Karena sistem komputer yang telah ada masih mampu untuk memberikan pelayanan yang memadai tanpa tambahan biaya lain, maka dari sisi ekonomi pun sistem komputer yang ada masih layak untuk digunakan sebagai basis dari aplikasi otentikasi nantinya.

- *Control* atau pengendalian adalah untuk mengetahui apakah sistem komputer yang telah ada menawarkan kontrol atau pengendalian untuk mengatasi kecurangan-kecurangan dan untuk menjamin keakuratan dan keamanan data. Dengan adanya penggunaan kata kunci (*password*) baik di lapisan iLO, BIOS, dan sistem operasi yang telah terpasang, maka pengendalian sistem komputer yang telah terpasang masih layak digunakan sebagai basis tempat aplikasi atau sistem otentikasi yang nantinya akan dipasang.
- *Efficiency* atau efisiensi adalah untuk mengetahui apakah sistem menggunakan secara maksimum sumber yang tersedia termasuk orang, waktu aliran *form*, meminimalkan penundaan proses. Dari sisi efisiensi pun sistem komputer yang sudah ada belum dimanfaatkan secara maksimum, sehingga dengan penambahan aplikasi atau sistem otentikasi nantinya diharapkan penggunaan *resource* atau sumber daya dari sistem komputer yang telah ada dapat lebih efisien karena mencapai nilai maksimum.
- *Services* atau pelayanan adalah untuk mengetahui apakah sistem komputer menyediakan layanan yang diinginkan dan handal pada siapa saja yang menginginkannya, dan apakah sistem fleksibel dan dapat dikembangkan. Melihat sistem komputer yang telah terpasang, kondisi pelayanan yang diberikan dari sistem komputer yang ada cukup handal dan fleksibel untuk dilakukan *upgrade* kedepannya. Sehingga dari sisi pelayanan sistem yang lama masih layak untuk digunakan sebagai tempat untuk dipasang aplikasi atau sistem otentikasi.

Setelah kita kaji kelayakan dari sisi operasional secara keseluruhan, dari setiap kerangka kerja PIECES yang menyatakan bahwa sistem otentikasi ini layak untuk dibangun dan dikembangkan dari tiap bagiannya, maka dari

faktor kelayakan operasional pun akan kita berikan nilai 9,0.

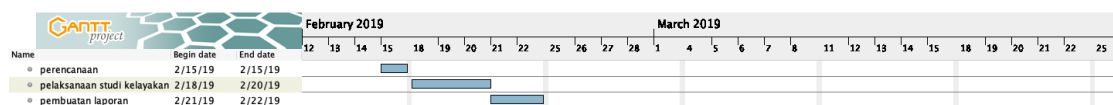
5. Menilai kelayakan Jadwal (*Schedule*)

Kelayakan jadwal digunakan untuk menentukan bahwa pengembangan sistem dapat dilakukan dalam batas waktu yang telah ditetapkan. Studi kelayakan sistem komputer direncanakan selesai dalam waktu maksimal ± 1 (satu) minggu.

Dalam studi kelayakan sistem komputer untuk pengembangan sistem otentikasi di Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah Kabupaten Brebes dilakukan dalam 3 (tiga) tahap, yaitu :

- (a) Tahap perencanaan
- (b) Tahap pelaksanaan studi kelayakan sistem komputer
- (c) Tahap pembuatan laporan

Grafik *Gantt* untuk tahapan-tahapan tersebut adalah seperti pada gambar 2 berikut ini :



Gambar 2: Grafik *Gantt*

Karena pengembangan diukur dalam jam, hari, minggu, dan bulan maka kesalahan perkiraan (*error estimation*) yang dibutuhkan untuk perancangan dan implementasi menjadi kecil. Maka nilainya 8.0

Dari keseluruhan faktor kelayakan TELOS tersebut, jumlah dari semua faktor-nya adalah

$$Total = 9 + 9 + 9 + 9 + 8$$

$$Total = 44$$

Total *score* dari kelayakan ini adalah :

$$score = 44/5 = 8,8$$

Artinya kelayakan sistem komputer yang akan digunakan untuk membangun sistem otentikasi di Badan Pengelolaan Pendataan, Keuangan dan Aset Daerah Kabupaten Brebes adalah LAYAK dengan resiko yang cukup rendah.

2 SARANA DAN PRASARANA YANG MELIPUTI PERANGKAT KERAS DAN PERANGKAT LUNAK YANG DIPERLUKAN

Kebutuhan akan sarana dan prasarana untuk jenis perangkat keras yaitu :

1. Peladen (*server*)

Peladen atau *server* berfungsi sebagai basis jalannya layanan-layanan aplikasi yang nantinya akan dibangun, nantinya tiap layanan aplikasi akan berjalan pada *container* Docker.

Secara keseluruhan, inti dari aplikasi ini akan berjalan pada 4 (empat) *container*, yang secara ringkas dapat dijelaskan seperti berikut :

- (a) *Container* pertama akan melayani *service* otentikasi, dimana halaman *login* akan berada pada layanan ini, termasuk pemberian token yang

digunakan oleh aplikasi klien untuk melakukan akses pada *resource server*.

- (b) *Container* kedua berupa layanan basis data yang secara langsung akan terhubung dengan *OAuth Service* atau layanan otentikasi pada *container* pertama dan *resource server*. Dalam layanan basis data ini tersimpan daftar pengguna, hak akses, serta daftar klien yang diijinkan melakukan koneksi ke beberapa *resource server* yang nantinya dibangun.
- (c) *Container* ketiga berupa layanan dari *resource server* yang nantinya digunakan untuk melakukan pengelolaan baik itu daftar pengguna atau daftar aplikasi klien yang diijinkan untuk terhubung. *Resource server* ini akan berbagi bersama layanan basis data pada *container* kedua dengan layanan *OAuth Server* pada *container* pertama.
- (d) *Container* keempat berupa layanan aplikasi klien yang berbentuk aplikasi *web*. Aplikasi klien ini akan menggunakan layanan dari *container* pertama, yaitu *OAuth service* sebagai layanan yang akan melakukan otentikasi terhadap pengguna, kemudian atas dasar hak dan kewenangan pengguna yang dapat melakukan akses ke pengelolaan daftar nama pengguna dan daftar aplikasi klien, mampu untuk melakukan akses ke *resource server*, apabila pengguna tidak memiliki hak dan kewenangan tersebut, maka layanan terhadap *resource server* secara sistematis akan ditolak.

Kebutuhan perangkat keras peladen (*server*) ini tidak perlu membeli karena nantinya akan menggunakan peladen (*server*) yang sudah ada sebagai peladen (*server*) aplikasi *web* yang sudah beroperasi.

2. Kartu Jaringan

Kartu jaringan digunakan atau terpasang pada peladen (*server*) agar peladen (*server*) dapat terhubung dengan jaringan, baik jaringan LAN (*Local Area Network*) maupun jaringan *internet*.

Kartu jaringan ini tidak perlu membeli karena sudah terpasang lebih dari 1 (satu) pada tiap peladen (*server*) yang dapat dikonfigurasi secara mandiri seluruhnya.

Melihat kondisi tersebut, kebutuhan perangkat keras secara keseluruhan sebenarnya tidak perlu mengeluarkan biaya tambahan lain karena perangkat-perangkat tersebut sudah tersedia dan cukup untuk mengakomodir berjalannya sistem informasi yang nantinya akan dibangun.

Kemudian, untuk sarana dan prasarana jenis perangkat lunak yang dibutuhkan yaitu :

1. Perangkat Lunak Sistem Operasi

Perangkat lunak sistem operasi digunakan sebagai dasar dari seluruh sistem yang akan berjalan di atasnya. Pemilihan sistem operasi ini pun harus dapat memenuhi kriteria stabil dalam melakukan tugasnya sebagai peladen (*server*) untuk rentang waktu 24 jam selama 7 hari berturut-turut.

Tentunya sistem operasi ini sudah terpasang baik pada peladen-peladen (*servers*) yang nantinya akan menjadi peladen (*server*) sistem basis data dan peladen (*server*) aplikasi *web*. Pada peladen (*server*) aplikasi *web* telah terpasang sistem operasi Ubuntu Server 14.04 yang tentu saja, untuk sistem operasi ini berlisensi gratis sehingga kebutuhan akan peladen (*server*) ini tidak perlu mengeluarkan biaya tambahan baik untuk lisensinya maupun pemasangannya.

2. Perangkat Lunak Sistem Basis Data

Perangkat lunak basis data digunakan sebagai sistem yang nantinya akan mengatur dan menyimpan data-data pengguna, data-data hak akses untuk tiap pengguna, serta daftar detail dari aplikasi klien yang diijinkan untuk melakukan akses terhadap *resource server*.

Pemilihan perangkat lunak basis data ini didasarkan pada lisensinya yang tersedia gratis, namun tidak mengurangi performa dan kehandalannya sebagai tempat simpanan data, sehingga pilihan jatuh pada sistem basis data Postgresql.

Sistem basis data ini sudah lebih dari 30 Tahun dikembangkan dengan lebih dari 400 kontributor, fiturnya pun tidak kalah dengan sistem basis data yang dikembangkan oleh industri-industri besar pada bidang sistem basis data.

Nantinya sistem basis data ini akan berada dalam salah satu *container* docker yang melakukan simpanan data untuk seluruh pengguna berikut hak aksesnya, basis data ini pun nantinya digunakan untuk menyimpan data atau informasi mengenai aplikasi klien yang mampu terhubung dan menggunakan sistem otentikasi yang akan dibangun.

Karena gratis, maka untuk menggunakan sistem basis data Postgresql ini tidak perlu mengeluarkan biaya tambahan untuk pengadaannya.

3. Perangkat Lunak Peladen *Web*

Perangkat lunak peladen (*server*) *web* digunakan untuk menjalankan atau menyediakan aplikasi *web* yang telah dibangun, yaitu sistem otentikasi di Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah Kabupaten Brebes.

Peladen (*server*) *web* ini nantinya akan ada 2 (dua) macam, yaitu yang berbentuk *web container* yang nantinya digunakan oleh *OAuth Service* dan

Resource Service, dan yang lain adalah *web server* seperti pada umumnya yang akan melayani aplikasi klien yang telah dibangun.

Untuk peladen (*server*) *web* yang berbentuk *web container* sebetulnya sudah ada dalam satu paket proyek yang dibangun menggunakan *framework* Spring Boot, nantinya paket proyek ini akan dimasukkan dalam sebuah *container* docker, sehingga proses *deployment* akan lebih mudah dengan model *Continuous Integration* dan *Continuous Deployment*.

Alasan kenapa dipilih peladen yang mendukung *servlet* atau *web container* karena memiliki beberapa kelebihan seperti berikut ini :

(a) Efisien dan baik dalam *performance*

Performance servlet dapat dikatakan efisien dan baik karena tidak ada proses pembuatan berulang untuk tiap *request* dari klien (*client*). Setiap *request* ditangani oleh proses *servlet container*. *Servlet* tidak dibuat dan dihancurkan secara berulang-ulang, melainkan tetap tersimpan pada memori untuk menangani *request* yang datang selanjutnya.

(b) *Powerful*

Servlet memiliki kemampuan yang lengkap antara lain mampu melakukan penanganan *request*, *session*, *cookie*, akses ke basis data dengan *Java Database Connection* (JDBC) dan *caching*, serta pustaka yang lengkap untuk pembuatan aplikasi *web*.

(c) Aman

Servlet memiliki fasilitas *security* atau keamanan yang baik dan merupakan bagian dari teknologi Java yang sudah dari asalnya didesain dengan *security* atau keamanan yang baik.

(d) Portabilitas

Teknologi Java *servlet* dapat dijalankan di berbagai *servlet container*, *application server*, maupun sistem operasi.

(e) Proses *development* yang lebih cepat

Dengan menggunakan *servlet*, kita dapat menggunakan pustaka Java yang lengkap dan menggunakan komponen yang sudah ada tanpa membangun dari awal kembali.

(f) Tangguh

Servlet merupakan teknologi Java yang memiliki penanganan *memory* yang baik dan *garbage collector* sehingga menjadikan aplikasi *web* yang tangguh dan stabil.

(g) Telah digunakan dan diakui di dunia

Servlet merupakan teknologi Java yang telah digunakan di berbagai belahan dunia. Dapat ditemukan berbagai komponen, solusi, dan dukungan yang ditawarkan baik secara gratis maupun komersial.

(h) Murah

Dikatakan murah karena JDK Java dapat diunduh secara gratis, begitupun dengan *servlet container*.

Pemilihan perangkat lunak peladen (*server*) *web* yang mendukung *servlet* ini pun tidak perlu mengeluarkan biaya, cukup menggunakan Apache Tomcat atau Jetty yang memberikan lisensi gratis dalam penggunaannya, dengan konfigurasi yang tepat, maka peladen (*server*) *web* Apache Tomcat ini cukup untuk melayani permintaan aplikasi dari beberapa klien sekaligus. Sehingga untuk pengadaan perangkat lunak peladen (*server*) *web* ini pun tidak memerlukan tambahan biaya, dan sudah ada dalam paket *framework* Spring Boot untuk kemudahan *debugging* dan *deployment*.

Perangkat lunak peladen (*server*) yang digunakan untuk aplikasi klien nantinya akan menggunakan Nginx yang juga tersedia secara gratis dan sudah teruji kehandalannya karena beberapa perusahaan teknologi informasi besar pun menggunakan Nginx sebagai *web server*-nya.

Ketiga peladen (*server*) ini akan ditempatkan pada masing-masing docker *container* untuk memberikan layanan sesuai dengan tugas dan fungsinya.

4. Perangkat Lunak Desain Aplikasi

Perangkat lunak desain aplikasi digunakan untuk membuat desain pembangunan aplikasi, mulai dari desain struktur basis data, desain *Unified Modeling Language* (UML) yang digunakan untuk membuat spesifikasi standar untuk mendokumentasikan, menspesifikan, dan membangun sistem perangkat lunak, desain UML sendiri dapat menggambarkan atau memodelkan diagram struktur aplikasi dan diagram sifat atau cara kerja aplikasi.

Perangkat lunak atau desain aplikasi ini pun tidak perlu mengeluarkan biaya tambahan, ada perangkat lunak untuk desain aplikasi yang gratis dan disediakan secara daring oleh <https://www.draw.io>. Aplikasi ini memiliki fasilitas yang cukup banyak dan bukan hanya untuk membuat desain UML saja, melainkan desain dari struktur basis data pun mampu dilakukan, kemudian desain jaringan komputer pun dapat dilakukan di aplikasi ini.

5. Perangkat Lunak *Integrated Development Environment* (IDE)

Perangkat lunak IDE ini digunakan untuk membangun aplikasi *web*, baik dari sisi tampilan tatap muka pengguna (*user interface*) maupun dari sisi kode. Pengujian *unit testing* pun dapat dilakukan dengan menggunakan perangkat lunak ini.

Beberapa pilihan untuk perangkat lunak IDE ini pun beragam dan banyak

yang menyediakan secara gratis. Sebagai contoh adalah Netbeans dan Eclipse. Untuk pengembangan aplikasi ini dipilih IntelliJ IDEA sebagai perangkat dalam membangun layanan *OAuth Service* dan *Resource Service* karena tersedianya beragam *plugins* sebagai pendukung dalam membangun sebuah aplikasi, sedangkan untuk membangun aplikasi klien yang digunakan untuk melakukan pengelolaan atau manajemen pengguna dan aplikasi klien lain, akan menggunakan Visual Studio Code, karena pada Visual Studio Code ini pun terdapat beragam *plugins* yang mampu mendukung dan mempermudah pengerjaan aplikasi klien nantinya.

Kedua aplikasi itu, baik IntelliJ IDEA dalam hal ini, yang digunakan adalah versi komunitas (*community edition*) dan Visual Studio Code tersedia secara gratis dan bebas diunduh dari masing-masing laman penyediaanya, sehingga untuk pengadaan perangkat lunak IDE ini tidak perlu mengeluarkan biaya tambahan.

6. Perangkat Lunak Docker

Perangkat lunak docker ini digunakan untuk membangun *container-container* kecil yang nantinya berjalan secara independen yang tiap *container*-nya akan menjalankan sebuah layanan (*service*) tertentu yang spesifik, contohnya, sebuah *container* hanya akan menjalankan layanan (*service*) basis data saja, sedangkan *container* yang lain hanya akan menjalankan layanan (*service*) (*OAuth Service*), keduanya akan berjalan secara independen dan paralel yang apabila ingin kedua layanan atau *container* ini saling terhubung, akan melalui sebuah sambungan yang didefinisikan secara khusus.

Bisa diistilahkan bahwa docker ini adalah bentuk virtualisasi yang lebih ringan, karena sebetulnya, didalam sebuah *container* akan terdapat sebuah sistem operasi yang berjalan, serta beberapa aplikasi pendukung yang tuju-

annya spesifik untuk menjalankan layanan (*service*) utamanya. Contohnya, saat ada *container* yang memiliki layanan (*service*) basis data Postgresql, maka disana akan terdapat sebuah sistem operasi, basis data Postgresql, dan beberapa pustaka pendukung agar basis data Postgresql dapat berjalan. Untuk penggunaan docker ini kita pun dapat memilih menggunakan versi *Enterprise* atau *community edition*, untuk membangun aplikasi ini, sebetulnya kebutuhan untuk menggunakan versi *community edition* sudah cukup, sehingga kita tidak memerlukan biaya tambahan lain untuk mendapatkannya.

7. Perangkat Lunak Git

Perangkat lunak Git digunakan untuk melakukan *Continuous Integration* dengan lebih sempurna. Fungsi dari perangkat lunak Git ini sebetulnya tergolong dalam aplikasi *Versioning Control* yang melakukan pencatatan atau *log* dari kondisi perubahan berkas atau kandar, sehingga saat melakukan perubahan-perubahan yang terjadi pada fitur aplikasi yang dibangun dapat dilihat historis perubahannya, dan apabila ada kesalahan implementasi fitur dan perlu untuk dikembalikan kondisi ke beberapa keadaan diwaktu sebelumnya, mampu dilakukan dengan mudah tanpa harus mengubah kondisi berkas saat ini.

Fungsi lain dari perangkat lunak Git ini sebetulnya adalah untuk kebutuhan kolaborasi, dimana 1 (satu) atau lebih pengkode (*coder*) dapat mengerjakan pekerjaan koding (*coding*) secara bersama-sama dan paralel, setelah pekerjaan satu sub-modul atau sebuah fitur selesai, hasil kodenya dapat digabung sehingga menjadi satu fitur yang lengkap dan dapat digunakan.

Penggunaan Git ini tidak terlepas dari sebuah peladen (*server*) Git yang digunakan untuk mengumpulkan seluruh hasil penetapan (*commit*) kode dari

masing-masing pengkode (*coder*), yang digunakan nantinya sebagai peladen (*server*) Git adalah Gitlab, karena disana sudah terdapat fitur *Continuous Deployment* (CD) yang berfungsi untuk melakukan instalasi atau pemasangan secara otomatis pada peladen (*server*) VPS milik unit kerja Badan Pengelolaan Pendapatan, Keuangan dan Aset Daerah.

Pada proses *Continuous Integration* dan *Continuous Deployment* (CI/CD) nantinya, dapat dilakukan konfigurasi fase penyelarasan kode yang dapat diatur sesuai kebutuhan, contohnya, pada saat kode dari tiap pengkode (*coder*) akan digabung, maka otomatis akan melakukan *unit test* untuk memastikan bahwa tidak ada kesalahan logika program per sub-unit kode terkecil sekaligus melakukan *integration test* untuk memastikan bahwa fitur baru atau pembersihan *bugs* yang dilakukan tidak bermasalah apabila sistem berada pada lingkungan yang mengharuskannya terhubung ke beberapa sumber daya (*resource*) seperti sistem basis data, jaringan, atau hal lain yang dibutuhkan sistem untuk melakukan kegiatan operasionalnya.

Penggunaan Git dapat diperoleh dan diunduh dengan gratis dari <https://git-scm.com/>, sedangkan Gitlab sendiri menyediakan layanannya dalam beberapa versi sesuai kebutuhan, dan fitur CI/CD yang kita butuhkan sudah ada pada versi gratis yang ditawarkan oleh layanan ini, sehingga tidak perlu mengeluarkan biaya untuk menggunakan layanan dari Gitlab ini.

8. Kotlin *Compiler*

Compiler ini digunakan untuk melakukan kompilasi kode yang ditulis dengan bahasa pemrograman Kotlin. Pemilihan bahasa pemrograman Kotlin ini sebagai dasar pembangunan aplikasi atau sistem Otentikasi karena mampu bersinergi dengan sistem yang dibangun menggunakan bahasa Java, dengan kata lain, karena *framework* yang digunakan adalah Spring Boot

yang dibangun menggunakan bahasa pemrograman Java, penggunaan bahasa pemrograman Kotlin dengan menggunakan *framework* Spring sangat memungkinkan dengan ditambah kehandalan dari bahasa pemrograman Kotlin yang mampu mendeteksi atau mengurangi timbulnya *Null Exception* pada aplikasi.

Kotlin *Compiler* sendiri tersedia secara gratis dan dapat diunduh dari laman <https://kotlinlang.org/>, sehingga penggunaannya tidak memerlukan biaya tambahan.

9. Maven

Maven digunakan sebagai *build-tools* untuk mempermudah pengelolaan proyek yang akan dibangun, untuk menggunakan berbagai macam pustaka seperti diatas, cukup dideklarasikan pada sebuah berkas konfigurasi Maven dengan nama `pom.xml`, untuk melakukan pemaketan (*packaging*) pun sangat mudah, nantinya dengan Maven, sebelum sebuah proyek didistribusikan atau di *deploy* ke peladen (*server*), akan dilakukan serangkaian tes (*unit test* dan *integration test*) untuk memastikan bahwa kode yang dibangun telah memenuhi kelayakan untuk digunakan.

3 SUMBER DAYA MANUSIA YANG TERLIBAT DALAM SISTEM KOMPUTER

Penentuan sistem komputer yang layak untuk digunakan tidak terlepas dari Sumber Daya Manusia, adapun Sumber Daya Manusia (SDM) yang terlibat dalam penentuan kelayakan sistem komputer yang akan digunakan sebagai peladen (*server*) dari sistem otentikasi yang akan dibangun adalah seorang Fungsional Pranata Komputer yang secara kemampuan memiliki kompetensi untuk melakukan

studi kelayakan sistem komputer ini.

4 WAKTU DAN BIAYA YANG DIBUTUHKAN DALAM PEMBUATAN / PENGEMBANGAN SISTEM APLIKASI SECARA MENYELURUH

Dari sisi biaya, maka tidak diperlukan pengeluaran biaya untuk pembuatan / pengembangan ini, karena sebetulnya seluruh perangkat keras sudah terpasang dan berjalan normal, dan dari pengadaan perangkat lunaknya pun semuanya menggunakan versi gratis.

Sedangkan dari waktu yang dibutuhkan, akan menghabiskan waktu sekitar 1 (satu) minggu yang tentu saja ini waktu yang tidak terlalu lama untuk menentukan kelayakan sebuah sistem komputer yang akan digunakan.