

## Pengembangan Sistem Informasi Verifikasi Keaslian Hasil Test Swab di Dunia Penerbangan (Studi Kasus: Kantor Kesehatan Pelabuhan Kelas 1 Soekarno-Hatta)

Nur Aisyah Asriani<sup>1</sup>, Fajar Pradana<sup>2</sup>, Yuita Arum Sari<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>aimma.riani@gmail.com, <sup>2</sup>fajar.p@ub.ac.id, <sup>3</sup>yuita@ub.ac.id

### Abstrak

Kantor Kesehatan Penerbangan (KKP) Kelas I Bandara Soekarno-Hatta merupakan instansi pemerintah yang salah satu tugasnya adalah mengurus proses validasi surat keterangan negatif COVID-19 (SKNC) yang dijadikan salah satu syarat melakukan perjalanan udara. Dalam perkembangannya, sempat marak terjadi pemalsuan SKNC tersebut, yang ternyata lolos dari proses validasi KKP yang masih dilakukan secara manual dan luring. Untuk mengatasi permasalahan tersebut, penulis mencoba memberikan solusi berupa Sistem Informasi Verifikasi Keaslian Hasil Test Swab yang melakukan validasi SKNC secara otomatis dan daring, agar penerbangan dapat dilaksanakan dengan lebih aman dari resiko penularan COVID-19. Pengembangan sistem dilakukan dengan menggunakan metode prototype yang terdiri dari beberapa tahapan diantaranya pengumpulan data, analisis dari kebutuhan, membangun prototype, evaluasi prototype, perancangan sistem, implementasi sistem, dan pengujian sistem. Berdasarkan iterasi didapatkan total 13 kebutuhan dengan 2 aktor, lab tes COVID-19 dan KKP. Implementasi dilakukan menggunakan bahasa pemrograman javascript dengan framework reactjs dan nodejs. Pengujian sistem yang dilakukan sebanyak empat jenis, yang terdiri dari pengujian unit dengan 3 sampel pengujian, pengujian integrasi yang menghasilkan hasil valid, pengujian validasi yang mendapatkan hasil valid dari kasus uji, dan pengujian non-fungsional berupa *usability tersting* yang dilakukan dengan menggunakan metode *SUS (System Usability Scale)* mendapatkan poin 73.46 dari 15 responden.

**Kata kunci:** COVID-19, system usability scale, reactJS, nodeJS, penerbangan

### Abstract

The Soekarno-Hatta Airport Class I Aviation Health Office (AHO) is a government institution tasked with validating COVID-19 negative test result (CNTR) documents, which is one of the prerequisites for boarding a flight. At one point, there have been forgeries made to such documents, which passed the manual and offline validation process done by AHO. To solve this problem, the writer try to devise an Information System for Verification of Swab Test Result Authenticity, which performs the validation automatically and online, so that flying can be safer from the risk of COVID-19 spread. The system was built by using the prototype development method, which consists of several stages, including data collection, requirements analysis, prototype building, prototype evaluation, system design, system implementation and system testing. Based on iterations done, there were 13 requirements identified from 2 actors, COVID-19 testing labs, AHO and system admin. Implementation was done using the javascript programming framework ReactJS and NodeJS. System testing was divided into four stages which consist of unit testing with 3 samples, integration testing to produce valid results, validation testing to obtain valid results from test cases, and non-functional testing in the form of usability testing using the System Usability Scale (SUS) method that obtained a point of 73,46 from 15 respondents.

**Keywords:** COVID-19, system usability scale, reactJS, nodeJS, flight

### 1. PENDAHULUAN

Pemerintah memberlakukan Pembatasan Sosial Berskala Besar (PSBB) dengan melarang

wisatawan mancanegara untuk berkunjung ke Indonesia sejak WHO menyatakan COVID-19 sebagai pandemic pada Bulan Maret 2020. Selain Indonesia, beberapa negara juga

melakukan penghentian sementara rute penerbangan internasional (Yandri, 2020). Penurunan yang terjadi selama 2020 sebanyak 0,56 juta penumpang pada bulan Maret untuk rute internasional dan untuk rute domestik penurunan mencapai 59,15% (BPS, 2020). Sehingga Pemerintah mulai menerapkan kebijakan untuk memulihkan sektor penerbangan. Untuk meningkatkan rasa aman dan mengurangi penularan, pemerintah mewajibkan calon penumpang menunjukkan hasil tes pcr atau antigen yang diterbitkan dalam kurun waktu 1 hingga 3 x 24 jam tergantung tujuan penerbangan (Garuda Indonesia, 2021). Permasalahan yang muncul setelah adanya kebijakan ini adalah banyaknya oknum yang memalsukan surat keterangan negatif covid sebagai syarat terbang. Pada awal Oktober 2020, Taiwan mencatat 132 kasus positif pada saat kedatangan pekerja Indonesia, 76 diantaranya membawa surat keterangan negatif COVID-19 (Straitstimes, 2020). Sehingga pada 30 November 2020, Taiwan mengumumkan larangan masuk sementara bagi negara di Asia Tenggara terhitung sejak 4 hingga 17 Desember. Namun, dalam waktu sepekan kenaikan kasus positif COVID-19 di Indonesia meningkat menjadi 6000 dalam waktu sepekan, maka Taiwan mempertahankan larangan sampai waktu yang belum ditentukan (Straittimes, 2020). Pada 22 Juni 2021, sebanyak 9 penumpang pesawat dinyatakan positif COVID-19 dalam perjalanan dari Bandara Internasional Juanda Surabaya menuju Bandara Internasional Supadio Pontianak dengan dua diantaranya menggunakan surat SKNC (tes PCR) palsu. Masalah ini sangat disayangkan dan memiliki resiko terjadinya penularan COVID-19 saat melakukan penerbangan.

Pada permasalahan tersebut maka penulis akan mengembangkan sebuah sistem untuk melakukan proses verifikasi surat keterangan negative COVID-19. Tujuan sistem ini dikembangkan agar dapat mempermudah proses verifikasi surat keterangan negative COVID-19 di dunia penerbangan, meminimalisir pemalsuan surat keterangan negative COVID-19 karena surat dilengkapi dengan *QRCode*. Pada pengembangan sistem verifikasi keaslian surat keterangan negatif COVID-19 dapat digunakan oleh pegawai faskes dengan menambahkan data pasien, mencetak surat keterangan negatif COVID-19, memperbaharui data pasien, dan hapus data pasien. Juga dapat digunakan oleh pegawai KKP dengan melakukan scan,

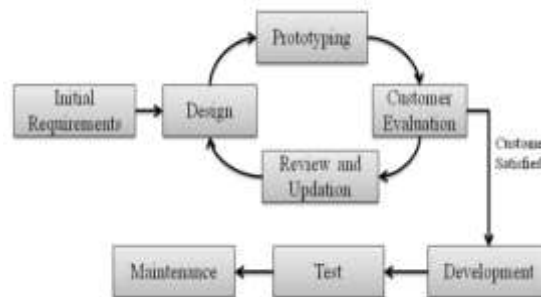
melakukan verifikasi keaslian surat keterangan negatif COVID-19, menambahkan data calon penumpang pesawat. Jadi dengan adanya sistem ini, pemalsuan surat keterangan negatif COVID-19 dapat dicegah dan meminimalisir pendataan calon penumpang pesawat secara manual.

Sistem ini akan dikembangkan menggunakan framework ReactJS dan nodeJS. Framework reactJS memberikan kemudahan dan kesederhanaa karena penulisan komponen melalui JSX sehingga dapat menyatukan javascript dan HTML. Selain itu, pengembangan sistem ini menggunakan model prototype. Salah satu alasan model prototype digunakan dalam pengembangan sistem ini karena pengembang dapat melakukan eksplorasi kebutuhan pengguna (Carr & Verner, 1997).

## 2. LANDASAN KEPUSTAKAAN

### 2.1 SDLC Prototyping

Penelitian ini menggunakan SDLC prototyping. Prototype termasuk model yang proses pendekatannya fokus pada kebutuhan pengguna pada sistem karena masukan pengguna penting pada proses pengembangan sistem (Isaias & Issa, 2015). SDLC prototype ditampilkan pada Gambar (Pressman, 2010).



Gambar 1. SDLC Prototyping

### 2.2 Class Diagram

*Class diagram* merupakan salah satu diagram yang menggambarkan sistem dari sudut setiap penjabaran kelas yang akan dibuat pada proses membangun sebuah sistem (Shalahudin & Rosa, 2008). *Class diagram* memiliki atribut dan juga konstrain yang saling menghubungkan antar objek. Class diagram juga memiliki relasi yang berbeda seperti asosiasi, dependensi, komposisi, agregasi, dan generalisasi (Hendini, 2016).

### 2.3 Sequence Diagram

*Sequence diagram* menjadi bagian dari UML yang digunakan untuk mendefinisikan setiap objek yang berada pada usecase yang diterima dan dikirim sesama objek (Shalahudin & Rosa, 2018). *Sequence*

diagram menggambarkan bagaimana objek berinteraksi didalam sistem seperti kelas atau komponen (Poranen, Erkki, & Nummenmaa, 2015).

**2.4 ReactJS**

ReactJS merupakan *library* javascript yang bersifat *open source* untuk membangun tampilan pengguna. Keunggulan dari reactJS diantaranya memberikan kecepatan, kemudahan, kesederhanaan dan skalabilitas (Tung Khuat, 2018).

**2.5 NodeJS**

NodeJS merupakan aplikasi server yang dibuat untuk proses pengembangan sistem berbasis web. Aplikasi ini dibuat dengan bahasa pemrograman javascript, menggunakan asynchronous dan basis *event*. (Iqbal, Husni, & Setiawan, 2012).

**2.6 Pengujian Blackbox**

Pengujian *blackbox* dilakukan untuk mendapatkan *error* dari fungsional pada proses implementasi kode program, kesalahan pada *user interface*, kesalahan pada struktur data, dan pada kinerja sistem (Pressman, 2010). Pengujian *blackbox* dilakukan sebagai pelengkap daripada pengujian *whitebox*, bukan sebagai pengganti. Pada pengujian ini, penulis menggunakan pengujian validasi.

**2.7 Pengujian Whitebox**

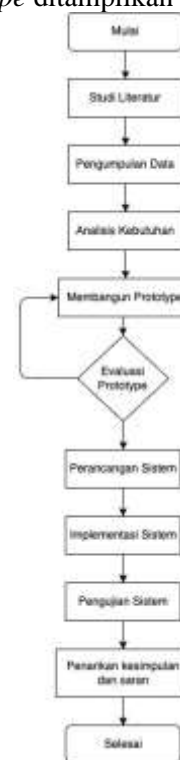
*Whitebox testing* merupakan teknik pengujian untuk merincikan logika internal dan struktur kode. Penguji perlu memahami keseluruhan kode program yang dikembangkan. Sehingga penguji menganalisis cara kerja internal sistem dan struktur sistem untuk menemukan (Khan & Khan, 2012). Pengujian *whitebox* dilakukan dengan pencarian jalur independen pada setiap kode program. Perancangan *flowgraph* mendapatkan jalur dengan tujuan supaya pengembang dapat menjamin semua jalur dapat ditelusuri (Pressman, 2010).

**2.8 Pengujian Usability**

Pengujian *usability* dilakukan dengan tujuan melihat kemudahan yang pengguna rasakan saat menggunakan sistem (Rubin & Chisnell, 2008). System Usability Scale (SUS) digunakan pada proses pengujian *usability* dengan memberikan 10 poin pertanyaan yang dikembangkan oleh John Brooke (Brooke, 2013).

**3. METODOLOGI**

Metodologi penelitian sistem verifikasi surat keterangan negatif COVID-19 menggunakan metode *prototype*. Tahapan metode *prototype* ditampilkan pda Gambar 2.



Gambar 2. Metode *Prototype*

Pada tahapan metodologi penelitian, penulis mencari studi pustaka yang digunakan untuk menjadi referensi pendukung pada penelitian dan pembelajaran agar dapat lebih memahami mengenai konsep yang akan diteliti dan membantu dalam proses analisis kebutuhan, perancangan sistem, implementasi sistem dan pengujian sistem. Pengumpulan data yang didapat oleh peneliti mengenai permasalahan yang terjadi perihal proses verifikasi keaslian surat keterangan negatif COVID-19 dengan menggunakan metode wawancara kepada pihak Klinik Salam Sarma bernama apt. Marsida, S. Farm dan petugas Kantor Kesehatan Kelas I Soekarno-Hatta bernama Ibu Fifi.

Perancangan sistem dilakukan dengan penulis menggunakan bahasa pemrograman javascript dengan *framework* reactJS dan nodeJS. Pada tahap ini penulis menggunakan perangkat keras berupa MacBook Pro 2017 dengan processor 2,3 GHz Dual-Core Intel Core i5, memori (RAM) 8 GB 2133 dan kartu grafis berupa Intel Iris Plus Graphics 640.

Proses pengujian dilakukan saat proses implementasi sudah selesai dan sesuai dengan kebutuhan pengguna. Proses pengujian ini menemukan error, mengetahui ketepatan kebutuhan fungsional dan non-fungsional. Penulis menggunakan *whitebox* dan *blackbox testing* sebagai teknik pengujian. Pada *whitebox testing* digunakan pada pengujian unit untuk melakukan pengujian pada kode program. Pada *blackbox testing* digunakan pada pengujian validasi untuk mengetahui sistem telah memenuhi standar fungsional dengan pembagian menjadi valid dan tidak valid. Dan pengujian non fungsional dilakukan dengan menggunakan *usability testing*.

Penarikan kesimpulan merupakan hasil dari penelitian yang dapat dilakukan pada akhir tahapan penelitian. Dan saran bertujuan memberikan masukan perbaikan kepada peneliti selanjutnya pada penelitian ini.

#### 4. ANALISIS KEBUTUHAN

##### 4.1 Identifikasi Aktor

Aktor yang menggunakan sistem ini dijelaskan pada Tabel 1.

Tabel 1. Identifikasi Aktor

Aktor	Deskripsi
Pegawai Lab/Klinik	Aktor yang melakukan tambah data pasien pada surat keterangan negative COVID-19.
Pegawai KKP	Aktor yang melakukan proses verifikasi validasi surat keterangan negative COVID-19 dan menambah data calon penumpang.

##### 4.2 Kebutuhan Fungsional

Kebutuhan fungsional perlu dilakukan saat mendefinisikan kebutuhan sistem. Proses identifikasi sistem bertujuan untuk menjelaskan kebutuhan pengguna. Sistem informasi keaslian surat keterangan negative COVID-19 memiliki 13 kebutuhan dengan salah satu kebutuhan dijelaskan pada Tabel 2.

Tabel 2. Kebutuhan Fungsional Tambah Data Pasien

Kode	Aktor	Nama Fungsi	Spesifikasi/ Deskripsi
SIV-F-L-003	Pegawai Lab	Tambah data pasien	Sistem dapat menambah dan menyimpan data terkait keterangan tes COVID-19

##### 4.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional fokus pada reaksi dan kemampuan pengguna dari sistem

yang dikembangkan. Kebutuhan non fungsional sistem ini yaitu *usability* dan dijelaskan pada Tabel 3.

Tabel 3. Kebutuhan Non Fungsional

Kode	Fitur	Deskripsi Kebutuhan
SIV-NF-001	<i>Usability</i>	Pengguna dapat menggunakan sistem dengan mudah.

#### 4.4 Use Case Scenario

*Use case scenario* memberikan penjelasan mengenai kebutuhan fungsional. Salah satu *use case scenario* dijelaskan pada Tabel 4.

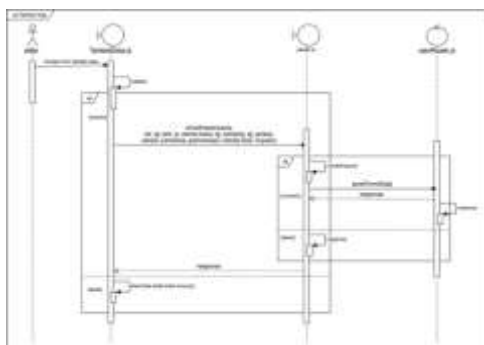
Tabel 4. Use Case Scenario

Nama Use Case	Tambah data pasien
<b>Aktor</b>	Pegawai lab/klinik
<b>Objective</b>	Aktor dapat menambahkan data pasien terkait keterangan hasil test COVID-19.
<b>Pre-condition</b>	Aktor berada pada halaman tambah data pasien.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>Aktor menekan tombol "Tambah Data".</li> <li>Sistem akan menampilkan halaman input data.</li> <li>Aktor mengisi seluruh field data calon penumpang.</li> <li>Aktor menekan tombol "Simpan".</li> <li>Sistem menyimpan data dalam database dan pesan "Data berhasil disimpan" ditampilkan oleh sistem .</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>Ketika aktor tidak mengisi seluruh atau Sebagian field data dan menekan tombol "simpan", maka sistem akan menampilkan pesan "Data tidak boleh kosong".</li> <li>Ketika aktor mengisi data yang sudah ada pada database dan mengisi seluruh field data, lalu aktor menekan tombol "Simpan", maka sistem menampilkan pesan "Data sudah terdaftar".</li> </ol>
<b>Post Condition</b>	Sistem berhasil menyimpan data dan pesan "Data Berhasil Disimpan" dtampilkan oleh sistem.

#### 5. PERANCANGAN DAN IMPLEMENTASI

##### 5.1 Perancangan Sequence Diagram

Proses kerja program pada pemenuhan kebutuhan sistem dijelaskan pada sequence diagram. Hasil identifikasi dari spesifikasi kebutuhan dan use case scenario sebagai objek dari *sequence diagram*. Sequence diagram digambarkan pada Gambar 3.



Gambar 3. Sequence Diagram

### 5.2 Perancangan Class Diagram

Gambaran mengenai elemen-elemen yang ada pada komponen serta hubungan antar relasi digambarkan pada perancangan *class diagram*. Tujuan dari perancangan *class diagram* agar sistem yang akan dikembangkan dapat dipetakan dengan jelas.

### 5.3 Perancangan Komponen

Pada bagian ini dijelaskan algoritma pseudocode. Pseudocode tambah data pasien ditampilkan pada Tabel 5.

Tabel 5. Pesudocode Tambah Data

```

Begin
onInput = fungsi untuk menampung
setiap variabel
onAddPasien = fungsi untuk
melakukan pengiriman data pasien
onAddHasil = fungsi untuk
melakukan pengiriman data Hasil
Seleksi kondisi if = jika inputan
kosong
Menampilkan alert "data tidak
boleh kosong"
Seleksi kondisi else = jika
inputan telah terisi
Simpan data
end
    
```

### 5.4 Perancangan Basis Data

Entity Relationship Diagram (ERD) digunakan dalam pemodelan proses perancangan basis data yang berisi gambaran dari beragam data yang dibutuhkan oleh sistem.

### 5.5 Perancangan Antarmuka

Perancangan antarmuka berisi rancangan-rancangan halaman antarmuka dari sistem verifikasi keaslian surat keterangan negatif COVID-19.



Gambar 4. Perancangan Antarmuka

### 5.6 Implementasi Basis Data

Proses implementasi basis data dari sistem ini menghasilkan 5 tabel basis data yang terdiri dari tabel lab, pasien, hasil, penumpang, dan penerbangan.

### 5.7 Implementasi Kode Program

Implementasi kode program adalah hasil dari implementasi yang dilakukan sesuai dengan kebutuhan fungsional dan perancangan sistem menjadi bentuk kode program.

### 5.8 Implementasi Antarmuka

Perancangan antarmuka yang telah dirancang sebelumnya dikembangkan dalam proses implementasi antarmuka. Antarmuka dijadikan sebagai perantara sistem dan pengguna sistem. Implementasi antarmuka daftar data ditampilkan pada Gambar 5.





Gambar 5. Implementasi Antarmuka

6, 10) merupakan poin poin pernyataan negatif. Untuk penilaian skor positif dilakukan dengan pengurangan 1 nilai dari respon pengguna, sedangkan penilaian skor negatif dilakukan dengan pengurangan 5 nilai dari respon pengguna. Kemudian menjumlahkan setiap skor *item* dan dikalikan dengan 2,5 untuk menghasilkan skor SUS (Brooke, 2013). Untuk pengujian SUS ditampilkan pada Tabel 4.

Tabel 4. Pengujian SUS Seluruh Responden

No	R	Nilai										Nilai (Jumlah * 2,5)
		1	2	3	4	5	6	7	8	9	10	
1	R1	4	3	4	3	2	3	3	2	4	2	75
2	R2	3	3	4	2	4	4	4	3	4	3	82.5
3	R3	3	3	2	2	3	3	3	2	3	1	62.5
4	R4	3	3	2	4	1	3	3	2	3	3	67.5
5	R5	4	3	3	4	4	2	3	3	3	2	77.5
6	R6	3	3	3	2	4	4	3	3	3	2	75
7	R7	2	3	2	2	3	3	3	3	3	2	62.5
8	R8	2	4	4	3	3	3	4	4	1	4	80
9	R9	3	4	4	3	4	3	4	4	4	3	80
10	R10	3	3	2	3	2	3	3	3	3	3	65
11	R11	3	3	3	4	3	2	2	3	3	4	67.5
12	R12	4	3	4	2	3	3	3	3	4	3	77.5
13	R13	3	2	3	3	3	1	2	4	4	1	65
14	R14	3	3	3	4	3	3	3	3	3	2	67.5
15	R15	3	4	4	4	4	4	4	4	4	4	97.5
73.46												
Good												

## 6. PENGUJIAN

### 6.1 Pengujian Whitebox

Pada pengujian *whitebox* menggunakan teknik pengujian unit. *Cyclomatic complexity* dan *basis path testing* digunakan pada proses pengujian *whitebox*. kedua metode ini dapat menghasilkan *independent path* yang digunakan untuk pengujian *test case*. *Independent path* dihasilkan dari proses *cyclomatic complexity*.

### 6.2 Pengujian Blackbox

Pada pengujian *blackbox testing* digunakan teknik pengujian validasi. Pengujian ini dilakukan untuk menguji fungsionalitas pada sistem. Pengujian validasi dilakukan berdasarkan *usecase scenario* yang telah dibuat.

### 6.3 Pengujian Usability

Pengujian *usability* dilakukan dengan melihat sejauh mana kemampuan pengguna dalam menggunakan produk dengan melakukan uji coba produk atau sistem untuk dilakukan pengamatan (Rubin & Chinsell, 2008). Pengukuran *usability* dilakukan dengan menggunakan pendekatan *system usability scale* (SUS) dengan memberikan 10 poin pernyataan pada SUS, dimana poin tersebut digunakan untuk memberikan penilaian sistem yang telah dibangun (Brooke, 2013). Pada poin dengan nomor ganjil (1, 3, 5, 7, 9) merupakan poin dengan pernyataan positif. Sedangkan poin dengan nomor genap (2, 4,

## 7. KESIMPULAN

Pada hasil analisis kebutuhan yang dilakukan pada pengembang sistem ini menghasilkan 12 kebutuhan fungsional dengan 2 aktor yaitu pegawai faskes dan pegawai KKP. Pada perancangan sistem terdapat perancangan *sequence diagram*, *class diagram*, basis data, perancangan antarmuka dan perancangan komponen. Pada perancangan basis data menggunakan ERD.

Pada tahapan implementasi menggunakan *framework* ReactJS dan nodeJS. Dalam tahap implementasi terdapat spesifikasi sistem, implementasi basis data, implementasi kode program, dan implementasi antarmuka.

Pada tahapan pengujian terdapat 3 metode pengujian diantaranya *whitebox testing*, *blackbox testing*, dan *usability testing*. Pada *whitebox testing* menggunakan Teknik pengujian unit dengan menguji 3 fungsi utama dalam sistem ini, yaitu pengujian unit tambah data pasien, perbaharui data pasien, dan verifikasi data penumpang. Dan dari keseluruhan hasil pengujian unit memiliki hasil valid. Pada pengujian *blackbox*, menguji 13 kebutuhan fungsional dengan teknik pengujian validasi dengan hasil valid. Dan pengujian *usability* menggunakan pendekatan System Usability Scale (SUS). Pada pengujian seluruh responden didapat nilai 73,46. Berdasarkan adjective ratings, sistem ini masuk pada kategori

'good' dan berdasarkan acceptability ranges masuk pada kategori *acceptable*.

## 8. SARAN

1. Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur mengirimkan surat hasil tes COVID-19 ke email pasien secara otomatis, sehingga dapat meminimalisir rusaknya surat cetak hasil tes COVID-19.
2. Berdasarkan skor dari pengujian usability yang didapatkan pada penelitian ini yaitu dengan nilai 73.46 dapat dilakukan evaluasi sehingga dapat meningkatkan nilai usability dari sistem verifikasi keaslian hasil tes swab di dunia penerbangan.

## 9. DAFTAR PUSTAKA

- Badan Pusat Statistika (2020), Jumlah Penumpang yang Berangkat pada Penerbangan Domestik di Bandara Utama Indonesia, 206-2020. <https://www.bps.go.id/linkTableDinamis/view/id/812>.
- Benony Walakula, Yandri. "Analisis Eksistensi Pariwisata Indonesia di Tengah Situasi Pandemi Corona Virus Disease (COVID-19)." *NOUMENA: Jurnal Ilmu Sosial Keagamaan* 1.1 (2020): 47-52.
- Brooke, J. (2013). SUS: a retrospective. *Journal of usability studies*, 8(2), 29-40.
- Carr, M., & Verner, J. (1997). Prototyping and software development approaches. *Department of Information Systems, City University of Hong Kong, Hong Kong*, 319-338.
- Iqbal C. R, Muhammad , Muchammad Husni, Hudan Studiawan, 2012. Implementasi Klien SIP Berbasis Web Menggunakan HTML5 dan Node.js. (<http://www.ejurnal.its.ac.id/index.php/teknik/article/view/643>). Diakses pada 120 Desember 2020.
- Khan, M. E., & Khan, F. (2012). A comparative study of white box, black box and grey box testing techniques. *Int. J. Adv. Comput. Sci. Appl*, 3(6).
- Khuat, T. (2018). Developing a frontend application using ReactJS and Redux.
- Poranen, T., Erkki, M., & Nummenmaa, J. (2015). How to Draw a Sequence Diagram, 4(Project 51528), 91–104.
- Pressman, R.S., 2010. *Software Engineering : A Practitioner's Approach*. Seventh Edition. New York : McGraw-Hill.
- Rubin, J., & Chisnell, D. (2008). How to plan, design, and conduct effective tests. *Handbook of usability testing*, 17(2), 348.
- Saias, P., & Issa, T. (2015). *High level models and methodologies for information systems* (pp. 21-33). New York, NW: Springer.
- Straittimes. (2020). Taiwan extends ban on Indonesian workers, citing coronavirus surge. *The Strait Times*, 16 Desember, 2020. <https://www.straitstimes.com/asia/east-asia/taiwan-extends-ban-on-indonesian-workers-citing-coronavirus-surge>.
- Shalahuddin, M. & Rosa, A. S., 2018. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Object*. Bandung: Informatika