Validasi Otomatis Dokumen Transkrip Nilai Mahasiswa Menggunakan Metoda Optical Character Recognition

Zulkarnaen Hatala [1], Ahmad Thariq [2], Muhammad Hudzaly[3], Muhammad Ikhwan Burhan[4]
[1,2] *Program Studi Teknik Informatika, Politeknik Negeri Ambon, Indonesia*
[3] *FakultasTeknik Industri, Universitas Diponegoro, Indonesia*
[4] *Institut Bisnis dan Keuangan Nitro, Makassar, Indonesia*

ARTICLE INFORMATION

Received: Maret 2023, 16
Revised: Maret 2023, 30
Available online: April 2023

KEYWORDS

Optical character recognition, Document image verification, Information retrieval

CORRESPONDENCE

Phone: +62 852 1630 5910
E-mail: dzulqarnaenhatala@gmail.com

A B S T R A C T

At the Ambon State Polytechnic, students' semester grade reports are still manually typed. This causes frequent typo errors which can result in the invalidity of the document, let alone incorrect grades, student identification numbers and many other label values. Here a java application has been implemented to detect these errors. This application is primarily intended for officials of the Head of Study Program, Head of the Department before signing and validating the report. Officials who legalize it will be greatly assisted because tedious validation work can be replaced by computers. The validation process is carried out by utilizing the optical character recognition technique from the open source library Tesseract-OCR. From the experimental results the verification process can be improved by using OCR specific on specific regions of interest (ROI) after using template matching method from OpenCV. The consideration of the Levehnstein distance in the comparison of label values against the reference database also improves the success rate of the algorithm. The database used has been tested for about 800 grade report documents, with successful verification result above 90%.

PENDAHULUAN

Pada negara dunia ke-3 atau negara berkembang, sebagian besar organisasi belum memiliki sistem informasi terkomputerisasi yang memadai (Sipe-Haesemeyer, 2005). Di Indonesia, organisasi kependidikan masih banyak yang menggunakan sistem manual baik sebagian maupun keseluruhan dalam melakukan proses bisnisnya. Sebagai contoh kita lihat di universitas dan politeknik pada proses pembuatan transkrip nilai akademik mahasiswa. Baik itu transkrip tahapan per semester maupun transkrip akhir secara keseluruhan. Transkrip nilai akademik mahasiswa adalah daftar nilai yang dicapai oleh seorang mahasiswa dalam periode atau tahapan semester sering disebut kartu hasil studi (KHS) atau juga daftar nilai sementara (DNS) atau transkrip semester. KHS atau DNS berisikan nilai untuk sekelompok mata kuliah dalam suatu semester. Persoalan yang kita amati adalah transkrip akademik ini ternyata masih diketik secara manual menggunakan aplikasi pengolah kata atau *spreadsheet* kemudian mencetaknya di kertas. Hal ini adalah kelemahan sistem informasi lembaga tersebut, di mana dns atau khs sebenarnya bisa dibuat secara otomatis. Di sini bisa terjadi salah pengetikan sehingga dokumen itu menjadi tidak valid lagi.

Isian transkrip memuat identitas mahasiswa, daftar mata kuliah, dan nilai yang diterima untuk setiap mata kuliah tersebut. Nilai-nilai ini serta informasi lain pada kertas yang tercetak mungkin mengalami kesalahan pengetikan. Kesalahan terjadi ketika juru ketik salah memasukkan data, di mana data tersebut bertentangan dengan nilai sebenarnya jika dibandingkan hasil proses evaluasi akademik sebelumnya seperti hasil ujian tengah dan akhir semester, hasil rapat evaluasi semester dan lain-lain. Pada sisi lain, dokumen yang tercetak tersebut akan legalkan dengan tandatangan pejabat berwenang jika dan hanya jika dokumen tersebut terbebas dari kesalahan. Biasanya penanda tangan yang berwenang memeriksa transkrip ini secara manual sebelum menandatangani kertas. Tetapi sebagai juru ketik, penandatangan juga manusia yang dapat melakukan kesalahan yang sama karena faktor kesalahan manusia (*human error*). Kesalahan manusia ini bisa terjadi jika melakukan suatu pekerjaan secara berulang-ulang misalnya dalam hal pengetikan ratusan transkrip secara manual. Cara manual ini bisa menimbulkan kelelahan dan kebosanan apalagi jika dilakukan secara terburu-buru dan diinginkan semua dokumen selesai di cetak dalam waktu yang sangat singkat(Yeow et al., 2014). Pada saat dokumen akan ditandatangani proses validasi terhadap ratusan dokumen juga akan menyebabkan efek yang sama bagi pejabat penandatangan dokumen. Dalam hal ini Koordinator Program Studi dan Ketua Jurusan bisa saja melakukan *human error* ketika menverifikasi ratusa dokumen secara manual.

Dalam makalah ini kami mengusulkan sebuah aplikasi komputer yang mendeteksi kesalahan pada dokumen tercetak. Aplikasi ini khusus diperuntukkan kepada penandatangan (*signer*) transkrip akademik nilai mahasiswa. Penandatangan akan dikonfirmasi apakah dokumen sudah bebas dari kesalahan, apakah informasi yang dicetak *valid* ataukah terkontaminasi dengan data yang salah. Jantung dari perangkat lunak ini menggunakan pemindai dokumen (scanner). Dokumen akan dipindai menjadi gambar atau informasi digital. Gambar ini kemudian diproses menggunakan teknik *optical character recognition* (ocr) (Fataicha et al., 2003; Lee et al., 2019; Yamakawa & Yoshiura, 2012)untuk mendapatkan informasi yang dicetak di atas kertas. Berdasarkan informasi ini, aplikasi akan menginformasikan otoritas tentang validitas informasi di atas kertas.

Aplikasi ini ditujukan untuk mendeteksi faktor kesalahan manusia (*human error factor*) yang mengakibatkan salah cetak dokumen transkrip nilai mahasiswa dan mencegah ditandatanganinya dokumen salah tersebut.

METODE PENELITIAN

Metoda penelitian yang digunakan adalah ujicoba terkontrol (*controlled experiments*) (Easterbrook et al., 2008). Tempat atau lokasi pengambilan data di Program Studi Teknik Informatika Politeknik Negeri Ambon. Dokumen yang digunakan sebagai data adalah transkrip nilai semester mahasiswa sebanyak kurang lebih 800 lembar. Contoh dokumen yang dipindai sebagaimana tertera dalam gambar 1. Dalam penelitian ini dibuatkan *software* menggunakan bahasa pemograman Java (Farrell, 2022)dengan tambahan pustaka (*library*) Tesseract(Smith, 2007) dan OpenCV(Gollapudi, 2019).

2.1 Algoritma Verifikasi

Secara umum cara kerja *software* adalah sebagaimana pada *flow chart* di gambar 1. Semua dokumen yang akan divalidasi dilakukan proses pemindaian menggunakan *flatbed scanner* agar diperoleh resolusi yang optimal yaitu 320 *dot per inch* (DPI). Hasil *scanning* ini adalah berupa file gambar dalam format *Joint Picture Expert Group* (JPEG). Setelah gambar dipindai maka dilakukan beberapa proses terhadap data digital tersebut(*preprocessing)*. Pada awalnya setelah di-*scan bitmap* akan dikonversi ke mode hitam dan putih kemudian dilakukan perataan (*deskewing*). Proses deskewing berguna jika pada saat melakukan pemindaian posisi kertas tidak rata terhadap *scanner*.

2.1.1 *Fullpage OCR*

Optical character recognition (ocr) adalah teknik untuk mendeteksi urutan karakter pada gambar dan mempunyai aplikasi yang sangat banyak sekali[3]–[5]. Setelah *preprocess*ing kini dilakukan konversi dari gambar (*image*) atau bitmap ke teks dengan metoda *optical character recognition* menggunakan *open source library* Tesseract-OCR[3]. OCR bisa dilakukan secara penuh pada halaman penuh (*fullpage)* atau bisa juga dilakukan pada area yang diinginkan *region of interest (*ROI*)* dari dokumen transkrip.

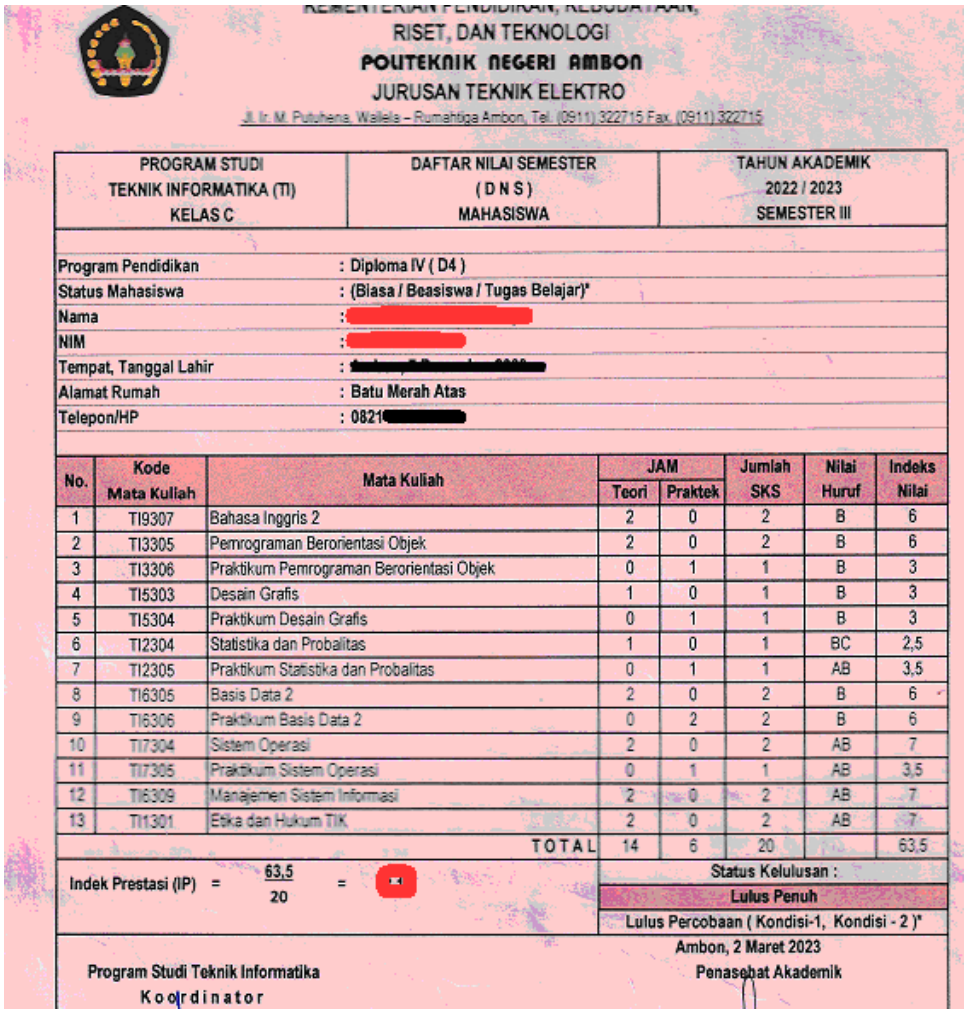
2.1.2 Pencarian label dan nilai *(Label value search)*

Keluaran dari prosess OCR yang dilakukan terhadap satu halaman penuh dokumen adalah berupa aliran teks (*text stream*). Pada *text sream* ini dicari nilai-nilai yang ingin diverifikasi dalam bentuk pasangan label (kata kunci) dan nilainya (*label value pair)*. Dalam verifikasi transkrip nilai di sini cuma digunakan 3 *label* yang ingin diverifikasi Jadi sebagaimana terlihat pada tabel 1. Aplikasi akan mencoba mencari 3 nilai tersebut dalam aliran teks dengan cara menemukan kata kunci NIM, NAMA dan IP kemudian mencari nilai dari ketiga kata kunci tersebut.

Aplikasi akan menverifikasi apakah 3 nilai ini sudah konsisten dengan basis data referensi yang ada. Pada kali referensi data adalah pada baris dan kolom yang bersesuaian di file Microsoft Excel. Jika 3 nilai untuk ke 3 label ini konsisten antara keluaran ***ocr*** dan data referensi di MS-Excel, maka dokumen dinyatakan valid dalam artian bisa ditandatangani.

Tabel 1. daftar pasangan label nilai

Label (kata kunci)	Contoh nilai	Keterangan
NIM	13170840001	Nomor induk mahasiswa
NAMA	Zulkarnaen Hatala	Nama Mahasiswa
Index Prestasi	3.98	Indeks prestasi mahasiswa



RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI AMBON
JURUSAN TEKNIK ELEKTRO
Jl. Ir. M. Putuhena, Wailela – Rumahtiga Ambon, Tel. (0911) 322715 Fax. (0911) 322715

PROGRAM STUDI TEKNIK INFORMATIKA (TI) KELAS C	DAFTAR NILAI SEMESTER (D N S) MAHASISWA	TAHUN AKADEMIK 2022 / 2023 SEMESTER III

Program Pendidikan : Diploma IV (D4)
Status Mahasiswa : (Biasa / Beasiswa / Tugas Belajar)*
Nama :
NIM :
Tempat, Tanggal Lahir :
Alamat Rumah : Batu Merah Atas
Telepon/HP : 0821

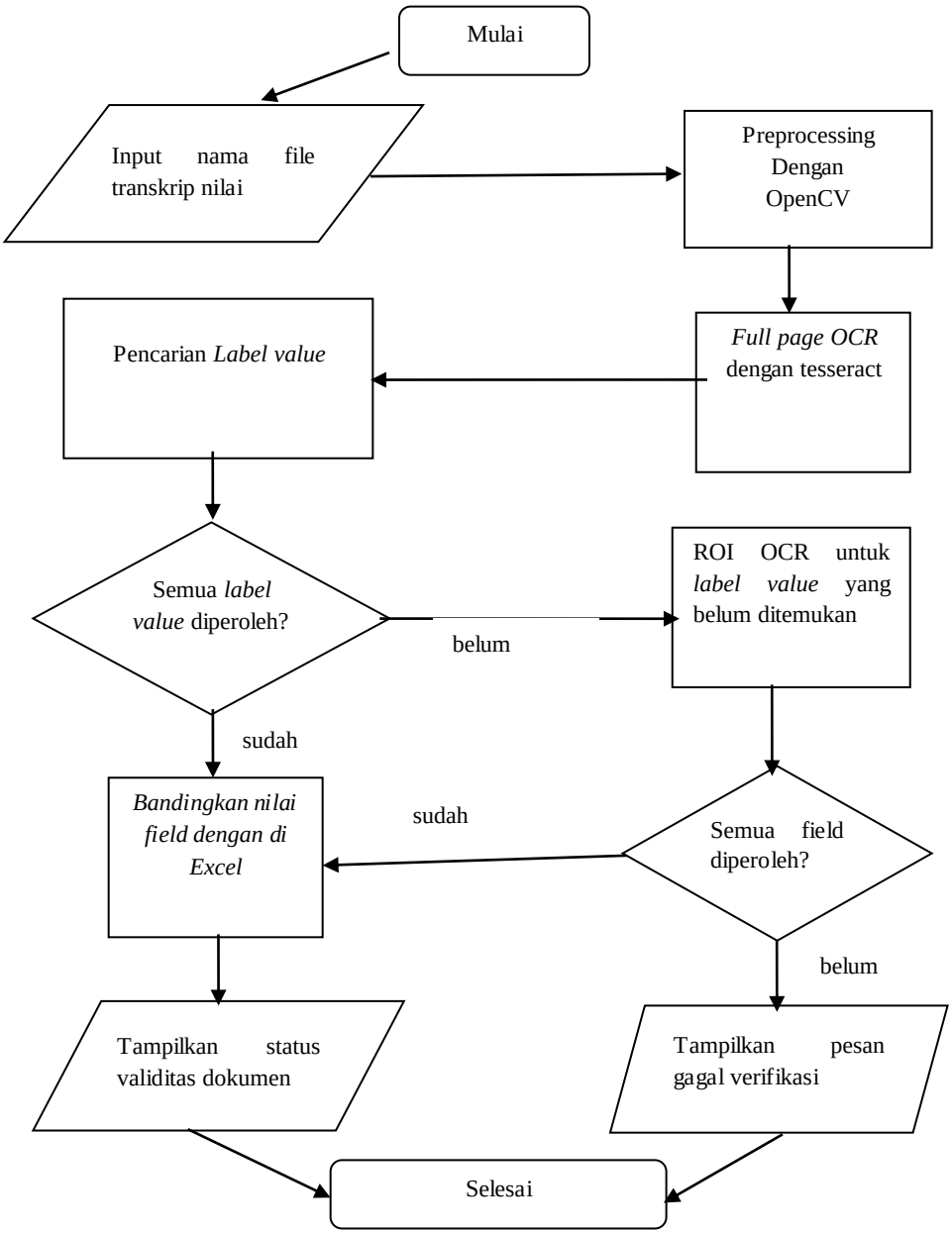
No.	Kode Mata Kuliah	Mata Kuliah	JAM Teori	JAM Praktek	Jumlah SKS	Nilai Huruf	Indeks Nilai
1	TI9307	Bahasa Inggris 2	2	0	2	B	6
2	TI3305	Pemrograman Berorientasi Objek	2	0	2	B	6
3	TI3306	Praktikum Pemrograman Berorientasi Objek	0	1	1	B	3
4	TI5303	Desain Grafis	1	0	1	B	3
5	TI5304	Praktikum Desain Grafis	0	1	1	B	3
6	TI2304	Statistika dan Probalitas	1	0	1	BC	2,5
7	TI2305	Praktikum Statistika dan Probalitas	0	1	1	AB	3,5
8	TI6305	Basis Data 2	2	0	2	B	6
9	TI6306	Praktikum Basis Data 2	0	2	2	B	6
10	TI7304	Sistem Operasi	2	0	2	AB	7
11	TI7305	Praktikum Sistem Operasi	0	1	1	AB	3,5
12	TI6309	Manajemen Sistem Informasi	2	0	2	AB	7
13	TI1301	Etika dan Hukum TIK	2	0	2	AB	7
		TOTAL	14	6	20		63,5

Indek Prestasi (IP) = 63,5 / 20 =

Status Kelulusan :
Lulus Penuh
Lulus Percobaan (Kondisi-1, Kondisi - 2)*

Ambon, 2 Maret 2023
Program Studi Teknik Informatika
Koordinator
Penasehat Akademik

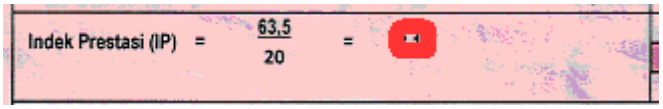
Gambar 1. Format dokumen transkrip nilai semester



Gambar 1: flowchart algoritma validasi dokumen transkrip nilai semester

2.1.1 *Region of Interest* (ROI) OCR dan *template matching*

Setelah melakukan *fullpage OCR*, bisa saja tidak ditemukan kata kunci (label) yang dinginkan pada *text stream* yang ada. Dalam kasus ini maka algoritma akan dilanjutkan lagi dengan proses OCR pada daerah (region) khusus saja. Sebagai contoh daerah khusus (*Region of Intererest* ROI) untuk mendapatkan label "Indeks Prestasi" adalah pada gambar 2.



Gambar 2. ROI OCR *Template* untuk label "Index Prestasi"

Tentunya ROI ini harus ditemukan terlebih dahulu menggunakan teknik *template matching*. Setelah ditemukan maka *ocr* hanya dilakukan terhadap ROI tersebut.

2.1.2 *Levehnstein (edit) distance*

Dalam penemuan label atau nilai seringkali kegagalan terjadi karena kesalahan relative kecil dibandingkan nilai referensi. Manusia akan sangat mudah sekali mendeteksi kegagalan palsu (*false alaram*) ini. Contohnya "Indek restasi" Cuma kehilangan 1 huruf 'P' saja untuk dikenali sebagai kata kunci ke 3 "Indek Prestasi". Dalam aplikasi ini akan menggunakan levehnstein distance (Srigiri & Saha, 2020) untuk mencegah *false alarm* ini terjadi.

HASIL DAN PEMBAHASAN

.Hasil utama dari penelitian ini adalah suatu aplikasi *open-source* berbahasa Java yang sudah diletakkan online di Github(Hatala, 2023) yang bisa diakses secara bebas oleh siapa saja.

3.1 Kekurangan Fullpage OCR

Sebagaimana telah disinggung sebelumnya bahwa hasil scan penuh per halaman doumen Setelah dilakukan OCR untuk keseluruhan halaman oleh Tesseract-OCR, ternyata tidak semua file JPEG bisa terkonversi secara akurat dengan tingkat kesalahan huruf nihi 0%. Apabila kesalahan terjadi pada label atau kata kunci maka *fullpage ocr* akan gagal menverifikasi dokumen semester dalam artian tidak bisa memberikan kesimpulan soal validitas dokumen.

1.2. Kesalahan-kesalahan manusia (*human factor error*)

Selama melakukan verifikasi terhadap kurang lebih 800 lembar dokumen transkrip nilai ditemukan banyak sekali kesalahan. Beberapa contoh kesalahan pengetikan yang sering dilakukan oleh juru ketik (*typist*):

1. Kesalahan kekurangan dan kelebihan angka (*deletion error)* pada NIM dengan angka mengulang
2. Kesalahan keterbalikan urutan huruf dan karakter (*switching error*)
3. Kesalahan pengetikan Indeks Prestasi

1.3. Java program

Hasil penelitian yang berupa aplikasi verifikasi transkrip nilai semester telah diletakkan di situs online. Salah satu *screenshot* dari aplikasi tersebut sebagaimana pada gambar 3. Pada gambar tersebut aplikasi mendeteksi bahwa “Indeks Prestasi” pada dokumen yang dicetak adalah 3.18 tidak sama dengan di referensi (MS Excel) 3.24.

KESIMPULAN

Dengan paparan di atas maka kita temukan bahwa verifikasi otomatis terhadap dokumen yang akan ditandatangani bisa dilaksanakan menggunakan *optical character recognition*. Penerapan ROI OCR dan *edit distance* bisa meningkatkan akurasi algoritma yang berarti semakin banyak dokumen yang bisa diverifikasi. Disimpulkan juga bahwa *software* ini bisa membantu *signer* untuk melakukan verifikasi dokumen yang akan ditandatangani sekaligus bertujuan mengurangi tingkat faktor kesalahan manusia (*human error*).

Gambar 3. *Screenshot* aplikasi verifikasi transkrip nilai

Kelemahan dari aplikasi pada saat ini adalah menerima output dari OCR (Tesseract) apa adanya. Belum memperbaiki melakukan pelatihan (*training*) terhadap *tesseract engine* untuk mengatasi permasalahan konversi teks yang timbul. Diharapkan dengan melakukan *training* maka tingkat kesuksesan verifikasi akan semakin bertambah mendekati angka 100%.

DAFTAR PUSTAKA

[1]. Easterbrook, S., Singer, J., Storey, M.-A., & Damian, D. (2008). Selecting empirical methods for software engineering research. *Guide to Advanced Empirical Software Engineering*, 285–311.

[2]. Farrell, J. (2022). *Java programming*. Cengage Learning.

[3]. Fataicha, Y., Cheriet, M., Nie, J. Y., & Suen, C. Y. (2003). Information Retrieval Based on OCR Errors in Scanned Documents. *2003 Conference on Computer Vision and Pattern Recognition Workshop*, 25–25. https://doi.org/10.1109/CVPRW.2003.10020

[4]. Gollapudi, S. (2019). *Learn computer vision using OpenCV*. Springer.

[5]. Hatala, Z. (2023). *Verifikator Transkrip Nilai Semester Otomatis* [Java]. https://github.com/dzhatala/scanned-document-verificator

[6]. Lee, Y., Song, J., & Won, Y. (2019). Improving personal information detection using OCR feature recognition rate. *The Journal of Supercomputing*, *75*(4), 1941–1952. https://doi.org/10.1007/s11227-018-2444-0

[7]. Sipe-Haesemeyer, M. A. (2005). Bringing the World Wide Web into Third World Countries: Integrating Technology Across the Globe. *Global Media Journal*, *4*(7).

[8]. Smith, R. (2007). An Overview of the Tesseract OCR Engine. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, *2*, 629–633. https://doi.org/10.1109/ICDAR.2007.4376991

[9]. Srigiri, S., & Saha, S. K. (2020). Spelling Correction of OCR-Generated Hindi Text Using Word Embedding and Levenshtein Distance. *Nanoelectronics, Circuits and Communication Systems: Proceeding of NCCS 2018*, 415–424.

[10]. Yamakawa, D., & Yoshiura, N. (2012). Applying Tesseract-OCR to detection of image spam mails. *2012 14th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 1–4.

[11]. Yeow, J. A., Ng, P. K., Tan, K. S., Chin, T. S., & Lim, W. Y. (2014). Effects of stress, repetition, fatigue and work environment on human error in manufacturing industries. *Journal of Applied Sciences*, *14*(24), 3464–3471.