



## *Deterministic Record Linkage* untuk Pembentukan Master Data Dosen

M. Miftakul Amin\*<sup>1</sup>, Atin Triwahyuni<sup>2</sup>

<sup>1</sup>Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya; Jl. Sriwijaya Negara, Bukit Lama, Kec. Ilir Bar. I, Kota Palembang, Sumatera Selatan 30128

<sup>2</sup>Jurusan Sistem Informasi, Institut Teknologi dan Bisnis PalComTech; Jl. Jend. Basuki Rachmat No.5, 20 Ilir D II, Kec. Ilir Tim. I, Kota Palembang, Sumatera Selatan 30151

\*Email Penulis Korespondensi: [miftakul\\_a@polsri.ac.id](mailto:miftakul_a@polsri.ac.id)

### **Abstrak**

*Sebuah sistem informasi menggunakan database sebagai tempat penyimpanan data secara permanen. Seiring bertambahnya data yang digunakan dalam aplikasi, maka terdapat penambahan data baik pada data master maupun data transaksi. Ketika sistem informasi berdiri sendiri, dan suatu saat akan digunakan basis data secara bersama-sama, maka dilakukan proses integrasi dari masing-masing sumber data. Tujuan dari penelitian ini adalah melakukan integrasi data dosen yang berasal dari bagian akademik dan pusat penelitian dan pengabdian kepada masyarakat Politeknik Negeri Sriwijaya. Metode yang digunakan dalam melakukan integrasi data adalah record linkage menggunakan teknik deterministic. Dengan mempertimbangkan bahwa masing-masing dataset memiliki unique identifier sebagai primary key yang digunakan. Hasil penelitian menunjukkan bahwa proses integrasi data menghasilkan master data dosen sebanyak 459 record, sebagai hasil dari integrasi data yang berasal dari sumber data yang berbeda.*

**Kata kunci**—*deterministic, record linkage, master data, dosen*

### **Abstract**

*An information system uses a database as a permanent data storage area. As more data is used in the application, there is additional data in both master data and transaction data. When the information system stands alone, and one day the database will be used together, the integration process is carried out from each data source. The purpose of this research is to integrate lecturer data from the academic department and the research and community service centre of Politeknik Negeri Sriwijaya. The method used in performing data integration is record linkage using deterministic techniques. By considering that each dataset has a unique identifier as the primary key used. The results showed that the data integration process produced 459 records of lecturer master data, as a result of data integration from different data sources.*

**Keywords**—*deterministic, record linkage, master data, lecturer*

## 1. PENDAHULUAN

Pengembangan sistem informasi seringkali melibatkan data yang berasal dari berbagai sumber dan memiliki representasi data yang berbeda-beda. Ketika data yang berasal dari sumber yang berbeda-beda ini akan diintegrasikan, maka akan ditemui data yang mengalami duplikasi, kesalahan ketik, dan diperoleh data yang kurang akurat. Seiring berjalannya waktu, maka volume data akan bertambah, baik pada data transaksi maupun data master yang digunakan dalam sebuah aplikasi. Dengan bertambahnya volume data dan kebutuhan integrasi data, maka pendekatan *record linkage* dapat digunakan untuk menjembatani sumber data yang berbeda sehingga diperoleh data yang akurat dan valid.

Di Politeknik Negeri Sriwijaya, terdapat beragam data yang bersifat parsial digunakan oleh beragam aplikasi. Sebagai contoh dalam sistem informasi akademik menggunakan data yang berbeda dengan sistem informasi penelitian dan pengabdian kepada masyarakat. Demikian juga, data yang digunakan oleh sistem informasi perpustakaan menggunakan sumber data yang berbeda dengan dua aplikasi sebelumnya. Pada titik tertentu, aplikasi yang digunakan pada masing-masing bagian diperlukan sinkronisasi, sehingga masing-masing memerlukan data yang *up-to-date*. Pada periode tertentu, terdapat penambahan data Dosen yang berasal dari rekrutmen pegawai dari bagian kepegawaian, dan terdapat kebutuhan untuk melakukan update data pada bagian akademik, bagian penelitian dan pengabdian kepada masyarakat, dan bagian perpustakaan. Sedangkan fakta tidak terintegrasinya masing-masing aplikasi dan sumber data yang berbeda, maka diperlukan tahapan untuk mendapatkan data yang akurat dan valid secara simultan pada masing-masing sistem informasi yang digunakan.

Pada domain *record linkage* ini memiliki beragam istilah yang mengacu pada konsep yang sama. Seperti yang disampaikan oleh Röchner et al. [1], terdapat istilah *record linkage*, *data deduplication*, *data matching*, dan *entity resolution*.

Menurut Soliman et al. [2], Sayers et al. [3], Kamat et al. [4], Xu et al. [5], Blake et al. [6], Clark et al. [7] *record linkage* sebagai sebuah tahapan dalam menyatukan informasi yang berasal dari sumber data yang berbeda dikelompokkan menjadi dua pendekatan, yaitu *deterministic* dan *probabilistic*. Pendekatan *deterministic* dilakukan manakala sumber data yang berbeda saling berbagai *unique identifier* yang sama. Sedangkan pendekatan *probabilistic* dilakukan jika tidak ada *unique identifier* yang sama antara masing-masing sumber data. Bahkan menurut Omar et al. [8] menambahkan satu pendekatan lagi disamping dua pendekatan sebelumnya, yaitu *machine learning record linkage* dalam klasifikasinya untuk meningkatkan hasil yang lebih optimal. Secara umum, terdapat 5 tahapan yang dilakukan dalam melakukan *record linkage* menurut Aghamohamadi et al. [9] yaitu *data pre-processing*, *indexing*, *record pair comparison*, *classification*, dan *evaluation*.

Penelitian tentang topik *record linkage* telah banyak dilakukan oleh para peneliti lain. Diantaranya Elezaj et al. [10] meneliti tentang *record linkage* menggunakan pendekatan *probabilistic* dan *data mining*. Teknik *record linkage* digunakan dalam tahapan *cleaning* yang merupakan bagian krusial dalam tahapan *data mining*. Penelitian ini menggunakan dua tahap penting, yaitu dengan melakukan ekstraksi *record* yang bersesuaian dari masing-masing sumber data, dan tahapan selanjutnya adalah mengeliminasi adanya duplikasi data yang ditemukan. Dalam upaya menemukan pasangan *record* yang bersesuaian dan mengurangi kompleksitas jumlah pasangan *record* yang akan dibandingkan menggunakan teknik *blocking*.

Demikian juga penelitian yang dilakukan oleh Churches et al. [11], melakukan penelitian tentang *record linkage* menggunakan pendekatan kombinasi antara *lexicon-based tokenisation* dan *probabilistic hidden Markov models (HMMs)*. Penelitian ini melakukan standarisasi pada data nama dan alamat, sebagai persiapan dalam melakukan proses *record linkage* lebih lanjut. Proses *record linkage* yang melibatkan data kematian akibat penyakit malaria dan *surveillance* telah dilakukan oleh Garcia et al. [12]. Algoritma *string similarity* yang digunakan adalah *Cosine* dan *Levenshtein*, untuk mengukur tingkat kemiripan antara nama pasien, tanggal lahir, dan nama ibu kandung. Penelitian ini telah berhasil mengidentifikasi sebanyak 216 kasus kematian atau sekitar 44.0% dari kedua data yang diintegrasikan.

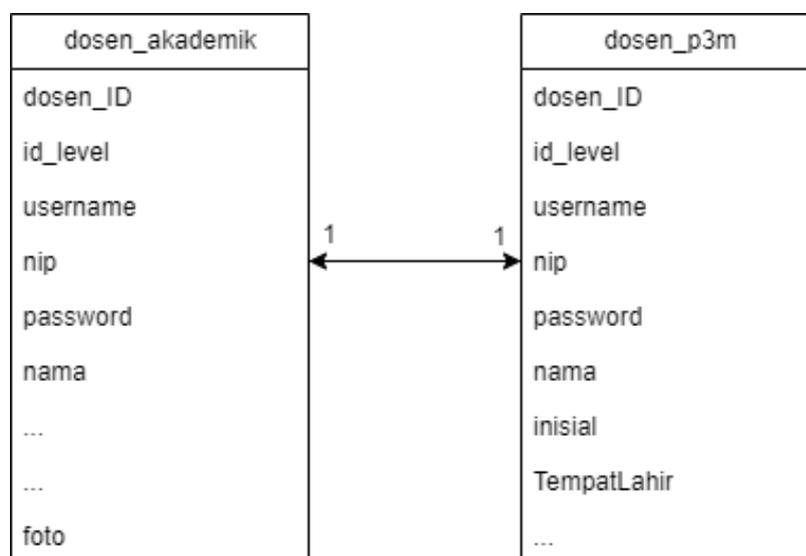
Lebih lanjut penelitian yang dilakukan oleh Vieira et al. [13], telah melakukan perbandingan beberapa pendekatan dalam *record linkage* diantaranya *Fellegi-Sunter*, *K-means*, *Bclust Hybrid*, *Bagging*, *ADA*, dan *SVM*. Dari sisi kecepatan eksekusi pendekatan *Fellegi-Sunter* lebih cepat dibandingkan pendekatan yang lainnya. Namun dari sisi hasil kinerja, pendekatan menggunakan supervised classifiers (*Bagging*, *ADA*, and *SVM*) menghasilkan kinerja yang lebih baik. Pendekatan *machine learning* juga dilakukan oleh Heydari et al. [14], dengan mengembangkan model *record linkage* untuk mengolah data kesehatan pasien yang diaplikasikan pada perangkat lunak *Apache Spark*. Hasil penelitian menunjukkan nilai evaluasi paling besar diperoleh dengan pendekatan *SVM* dengan nilai Recall sebesar 0.9992. sedangkan pendekatan menggunakan *Regression* mendapatkan nilai terbaik pada Accuracy sebesar 0.9627.

Penelitian yang dilakukan ini bertujuan untuk mengembangkan sebuah model integrasi data menggunakan teknik *record linkage* menggunakan pendekatan *deterministic record linkage* dengan mempertimbangkan bahwa masing-masing aplikasi sebagai sumber data yang mengolah data Dosen memiliki *unique identifier* sebagai *primary key*. Dengan adanya *primary key* ini, maka proses pencocokkan dan deteksi duplikasi dapat dilakukan dengan relatif lebih mudah.

## 2. METODE PENELITIAN

### 2.1 Dataset

Penelitian ini menggunakan dua buah dataset, yaitu dataset yang berasal dari bagian akademik dan bagian penelitian dan pengabdian kepada masyarakat (P3M). Penelitian ini menggunakan Dataset dari penelitian yang telah dilakukan oleh Amin et al. [15]. Gambar 1 memperlihatkan struktur dataset dan sumber datanya. Pada struktur data untuk tabel dosen yang berasal dari akademik sebanyak 36 kolom, dengan jumlah record sebanyak 444 buah. Sedangkan data dosen yang berasal dari p3m sebanyak 37 kolom, dengan jumlah record sebanyak 443 buah.



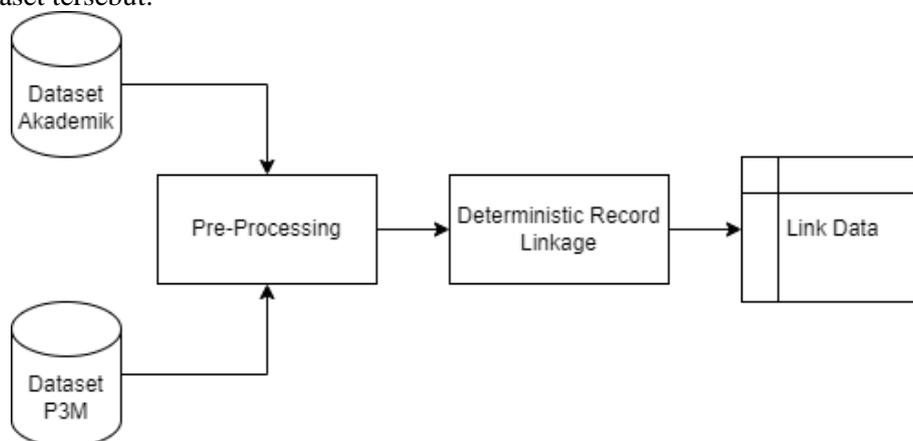
Gambar 1 Struktur Tabel Dataset

Pada Gambar 1 tampak relasi antara kedua buah tabel menggunakan nomor induk pegawai (*nip*) sebagai kolom penghubung antara kedua tabel tersebut dengan relasi *one to one* yang berarti bahwa pada kedua tabel tersebut hanya terdapat 1 *instance* *nip*.

### 2.2 Kerangka Kerja Model

Gambar 2 memperlihatkan kerangka kerja model yang dikembangkan. Setelah 2 buah dataset diperoleh, selanjutnya dilakukan tahapan *pre-processing* data pada kedua buah tabel tersebut. Tahapan *pre-processing* ini dilakukan dengan mengubah huruf pada *record* menjadi huruf kecil semua, dan mengeliminasi adanya *double white space* pada informasi yang terkandung

dalam kedua buah dataset. Selanjutnya dilakukan tahapan *deterministic record linkage* dengan menggunakan kolom nip sebagai penentu integrasi data. Setelah tahapan *deterministic record linkage* dilakukan, maka akan diperoleh dataset baru sebagai hasil proses link data antara kedua buah dataset tersebut.



Gambar 2 Kerangka Kerja Model

#### 2. 4 Pseudocode Levenshtein Distance

*Levenshtein Distance* adalah metrik yang digunakan untuk mengukur seberapa berbeda dua string teks dalam hal jumlah operasi (penyisipan, penghapusan, atau penggantian satu karakter) yang diperlukan untuk mentransformasikan satu string menjadi string lainnya. Jarak Levenshtein sering digunakan dalam bidang seperti pemrosesan bahasa alami, koreksi ejaan, dan pencocokan string [16]. Hasil dari pengukuran tingkat kemiripan akan menghasilkan nilai sebagai berikut:

- Nilai 0: Ini menunjukkan bahwa kedua string tersebut identik atau sama persis. Tidak ada operasi yang diperlukan untuk mentransformasikan satu string menjadi string lainnya, karena keduanya sudah sama.
- Nilai 1: Ini menunjukkan bahwa kedua string tersebut sangat mirip. Satu operasi penyuntingan saja (penyisipan, penghapusan, atau penggantian satu karakter) diperlukan untuk mentransformasikan satu string menjadi string lainnya.
- Nilai Lebih Besar dari 1: Semakin besar nilai Levenshtein Distance, semakin berbeda kedua string tersebut. Nilai ini menunjukkan bahwa lebih banyak operasi penyuntingan yang diperlukan untuk mentransformasikan satu string menjadi string lainnya.

```

FUNCTION LevenshteinDistance(s1, s2)

  # Inisialisasi matriks untuk menyimpan jarak
  m = CREATE MATRIX(length(s1) + 1, length(s2) + 1)

  # Inisialisasi baris dan kolom pertama matriks
  FOR i FROM 0 TO length(s1)
    m[i][0] = i
  FOR j FROM 0 TO length(s2)
    m[0][j] = j

  # Isi sisa matriks lainnya
  FOR i FROM 1 TO length(s1)
    FOR j FROM 1 TO length(s2)
      IF s1[i] == s2[j] THEN
        cost = 0
      ELSE
        cost = 1
      m[i][j] = minimum(m[i - 1][j] + 1, // Penghapusan
                        m[i][j - 1] + 1, // Penyisipan
                        m[i - 1][j - 1] + cost) // Substitusi

  # Kembalikan jarak di sudut kanan bawah matriks
  RETURN m[length(s1)][length(s2)]
  
```

## 2. 4 Pseudocode Deterministic Record Linkage

Pada tahapan *deterministic record linkage* ini menggunakan nomor induk pegawai (nip) pada kedua buah tabel dengan mengimplementasikannya dalam bentuk fungsi yang diberi nama *match\_key* (). Jika jarak antara kedua nip bernilai 0, maka akan mengembalikan nilai logika *True*, demikian juga sebaliknya. Berikut merupakan *pseudocode* yang digunakan untuk mengimplementasikan *deterministic record linkage*.

```
FUNCTION match key(nip akademik, nip p3m)

# Periksa apakah nip identik (jarak Levenshtein adalah 0)
IF LevenshteinDistance(nip_akademik, nip_p3m) == 0 THEN
  RETURN True

# Periksa apakah nip akademik merupakan substring dari nip p3m atau sebaliknya
IF nip_akademik IN nip_p3m OR nip_p3m IN nip_akademik THEN
  RETURN True

# Nama pengguna tidak cocok
RETURN False
```

## 3. HASIL DAN PEMBAHASAN

### 3.1 Hasil

Penelitian ini menggunakan tahapan penelitian yang mengacu pada kerangka kerja model seperti dijelaskan pada Gambar 2. Berikut merupakan pemrosesan yang dilakukan dalam *deterministic record linkage*.

#### 3.1.1 Loading Dataset

Penelitian ini menggunakan Bahasa pemrograman python untuk melakukan setiap tahapan yang ada dalam kerangka kerja penelitian. Pada tahapan ini dua buah dataset akan dimuat dalam memori kerja untuk pemrosesan lebih lanjut. Untuk melakukan *loading dataset* menggunakan segmen kode program berikut.

```
1 # dataset 1 db: akademik
2 # dataset 2 db: p3m
3
4 df_akademik = pd.read_csv('dataset/dosen_akademik.csv', skip_blank_lines=True).astype(str)
5 df_p3m = pd.read_csv('dataset/dosen_p3m.csv', skip_blank_lines=True).astype(str)
```

Tidak semua kolom digunakan dalam pemrosesan lebih lanjut. Sehingga beberapa kolom yang sesuai akan dipilih sebagai fitur utama. Hal ini untuk memudahkan melakukan investigasi terhadap pengolahan datanya. Untuk melihat struktur data daring masing-masing dataset dapat menggunakan segmen kode program sebagai berikut.

```
1 # menampilkan kolom tertentu saja
2
3 df_akademik = df_akademik[['username', 'nama', 'TempatLahir', 'TanggalLahir', 'Kelamin_ID']]
4 df_akademik
```

```
1 # menampilkan kolom tertentu saja
2
3 df_p3m = df_p3m[['username', 'nama', 'TempatLahir', 'TanggalLahir', 'Kelamin_ID']]
4 df_p3m
```

#### 3.1.2 Pre-Processing Data

Tahapan *pre-processing* ini dilakukan untuk mengubah informasi yang terdapat pada setiap *record* menjadi huruf kecil semua, kemudian menghapus jika terdapat informasi berupa spasi lebih dari satu, baik berada di awal, di tengah, maupun di akhir kalimat. Perhatikan segmen kode program berikut.

```

1 def casefolding(text):
2     text = text.lower() # 1. merubah bentuk menjadi lower case
3     text = text.replace('\s+', ' ') # 2. menghapus multiple whitespace
4     text = text.strip()
5     return text

```

Proses *pre-processing* ini dikemas dalam sebuah fungsi, sehingga memudahkan dalam implementasi kode program yang dapat dipanggil atau digunakan secara berulang. Fungsi *casefolding* () ini selanjutnya diimplementasikan pada masing-masing dataset, yang diaplikasikan pada kolom nama dan tempat lahir, seperti diperlihatkan pada segmen kode program berikut.

```

1 df_akademik['nama'] = df_akademik['nama'].apply(casefolding)
2 df_akademik['TempatLahir'] = df_akademik['TempatLahir'].apply(casefolding)
3 df_akademik

```

```

1 df_p3m['nama'] = df_p3m['nama'].apply(casefolding)
2 df_p3m['TempatLahir'] = df_p3m['TempatLahir'].apply(casefolding)
3 df_p3m

```

### 3.1.3 Tahapan Deterministic Record Linkage

Proses *deterministic* ini menggunakan nomor induk pegawai (nip) sebagai informasi yang digunakan secara bersama-sama oleh masing-masing *dataset* untuk acuan integrasi data. Proses yang dilakukan adalah mengemasnya dalam bentuk fungsi yang diberi nama *match\_key* (). Fungsi ini membutuhkan 2 buah parameter dalam eksekusinya, yaitu nip yang berasal dari dataset akademik dan dataset p3m. fungsi *match\_key* () ini akan mengembalikan nilai True jika pengukuran tingkat kemiripan (*similarity*) dari dua buah parameter menghasilkan nilai 0, demikian sebaliknya akan mengembalikan nilai False jika dua buah parameter tersebut bernilai 1 atau lebih besar dari 1.

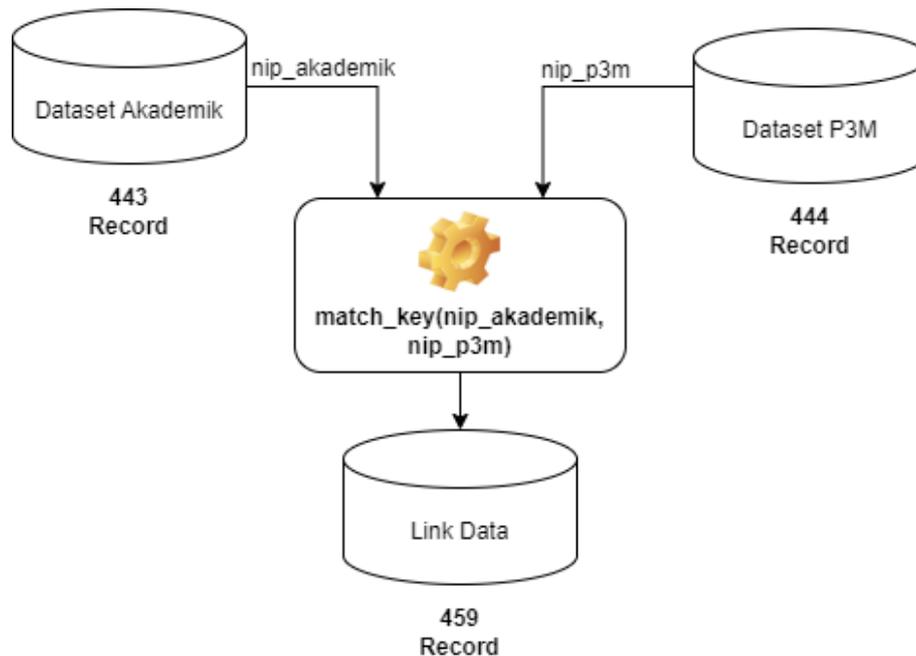
```

1 # This function returns true if 2 unique identifier are a match
2 def match_key(nip_akademik, nip_p3m):
3     if textdistance.levenshtein(nip_akademik, nip_p3m) == 0:
4         return True
5     if (nip_akademik in nip_p3m) or (nip_p3m in nip_akademik):
6         return True
7     return False

```

### 3.2 Pembahasan

Hasil penelitian menunjukkan bahwa proses integrasi telah dapat dilakukan dari dua buah *dataset*. Gambar 3 memperlihatkan bagaimana mekanisme dari tahapan *deterministic record linkage*. Fungsi *match\_key* () memerlukan 2 buah parameter dalam eksekusinya, yaitu nip yang berasal dari dataset akademik dan dataset P3M. Proses *deterministic record linkage* diterjemahkan secara eksplisit dengan mengemas sebuah fungsi yang bernama *match\_key* (). Pada prinsipnya, kedua dataset akan dilakukan proses penggabungan (*merge*) sehingga dihasilkan sebuah dataset baru. Proses penggabungan ini akan dilakukan jika pada kedua dataset terdapat nomor induk pegawai (NIP) yang sama. *Record* pada kedua *dataset* yang tidak memiliki informasi NIP yang bersesuaian akan diabaikan, dan dianggap sebagai informasi yang berbeda. Oleh karena itu jika dianalogikan sebagai sebuah himpunan, maka akan terbentuk dataset baru yang terdiri dari anggota himpunan yang beririsan ditambah dengan anggota himpunan dari kedua himpunan yang bukan irisan. Sehingga dari 443 *record* dan 444 *record* dari kedua dataset, akan diperoleh hasil berupa *record* data baru sebanyak 459 *record*.



Gambar 3 Mekanisme Tahapan *Deterministic Record Linkage*

#### 4. KESIMPULAN

Penelitian ini telah berhasil mengembangkan sebuah *model record linkage* untuk melakukan integrasi data Dosen yang berasal dari bagian akademik dan pusat penelitian dan pengabdian kepada masyarakat. Informasi nomor induk pegawai (NIP) digunakan sebagai *unique identifier* yang digunakan secara bersama-sama antara masing-masing sumber data. Hasil penelitian menunjukkan bahwa model yang dikembangkan telah mampu mengintegrasikan kedua sumber data menjadi data tunggal, dengan mengeliminasi data yang sama atau duplikasi pada saat *record linkage* dilakukan. Dari dua sumber data masing-masing memiliki 444 record dan 443 record, sehingga terbentuk sejumlah 459 record data baru sebagai hasil integrasi.

#### 5. SARAN

Penelitian ini telah berhasil melakukan integrasi data, namun hasil dari proses integrasi dari dua buah dataset masih terdapat duplikasi di dalamnya. Oleh karena itu sebagai penelitian lanjutan dapat dilakukan proses untuk mengeliminasi adanya duplikasi data yang terbentuk, sehingga menghasilkan data yang berkualitas.

#### UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tim Redaksi Jurnal Teknik Politeknik Negeri Sriwijaya yang telah memberi kesempatan, sehingga artikel ilmiah ini dapat diterbitkan.

#### DAFTAR PUSTAKA

- [1] P. Röchner and F. Rothlauf, "Using machine learning to link electronic health records in cancer registries: On the tradeoff between linkage quality and manual effort," *Int. J. Med. Inform.*, vol. 185, no. October 2023, p. 105387, 2024, doi: 10.1016/j.ijmedinf.2024.105387.
- [2] A. Soliman and S. Rajasekaran, "FIRLA: a Fast Incremental Record Linkage Algorithm," *J. Biomed. Inform.*, vol. 130, no. November 2021, p. 104094, 2022, doi: 10.1016/j.jbi.2022.104094.

- 
- [3] A. Sayers, Y. Ben-Shlomo, A. W. Blom, and F. Steele, "Probabilistic record linkage," *Int. J. Epidemiol.*, vol. 45, no. 3, pp. 954–964, 2016, doi: 10.1093/ije/dyv322.
- [4] G. Kamat, M. Shan, and R. Gutman, "Bayesian record linkage with variables in one file," *Stat. Med.*, vol. 42, no. 27, pp. 4931–4951, 2023, doi: 10.1002/sim.9894.
- [5] S. Xu, M. Zheng, and X. Li, "String Comparators for Chinese-Characters-Based Record Linkages," *IEEE Access*, vol. 9, pp. 3735–3743, 2021, doi: 10.1109/ACCESS.2020.3047927.
- [6] H. A. Blake, L. D. Sharples, K. Harron, J. H. van der Meulen, and K. Walker, "Probabilistic linkage without personal information successfully linked national clinical datasets," *J. Clin. Epidemiol.*, vol. 136, pp. 136–145, 2021, doi: 10.1016/j.jclinepi.2021.04.015.
- [7] J. Website, "International Journal of data : is it possible without patient identifiers ? A case study from the UK," *Int. J. Popul. Data Sci.*, vol. 4, no. August, pp. 1–12, 2019.
- [8] Z. Ali Omar, Z. H. Zamzuri, N. Mohd Ariff, and M. A. Abu Bakar, "Training Data Selection for Record Linkage Classification," *Symmetry (Basel)*, vol. 15, no. 5, 2023, doi: 10.3390/sym15051060.
- [9] Z. Rezaei Ghahroodi and Z. Aghamohamadi, "Record Linkage with Machine Learning Methods," *J. Stat. Sci.*, vol. 16, no. 1, pp. 1–24, 2022, doi: 10.52547/jss.16.1.1.
- [10] O. Elezaj and G. Tuxhari, "Record Linkage using Probabilistic Methods and Data Mining Techniques," *Mediterr. J. Soc. Sci.*, vol. 8, no. 3, pp. 203–207, 2017, doi: 10.5901/mjss.2017.v8n3p203.
- [11] T. Churches, P. Christen, K. Lim, and J. X. Zhu, "Preparation of name and address data for record linkage using hidden Markov models," *BMC Med. Inform. Decis. Mak.*, vol. 2, no. 1, 2002, doi: 10.1186/1472-6947-2-9.
- [12] K. K. S. Garcia *et al.*, "Record Linkage for Malaria Deaths Data Recovery and Surveillance in Brazil," *Trop. Med. Infect. Dis.*, vol. 8, no. 12, 2023, doi: 10.3390/tropicalmed8120519.
- [13] M. A. A. Z. Vieira and K. Louise e Silva, "Comparison of record linkage methods," *Rev. Gestão e Secr. (Management Adm. Prof. Rev.)*, vol. 14, no. 5, pp. 7999–8004, 2023, doi: 10.7769/gesec.v14i5.2171.
- [14] M. Heydari and R. Sarshar, "Distributed Record Linkage in Healthcare Data with Apache Spark," in *Conference: Artificial Intelligence and Software Engineering Conference (AISoft)*, 2023, no. October.
- [15] M. M. Amin, A. Sutrisman, and Y. Dwitayanti, "Master Data Management using Record Linkage Toolkit for Integrating Lecturer Master Data," *E3S Web Conf.*, vol. 448, 2023, doi: 10.1051/e3sconf/202344802062.
- [16] N. Anggraini and M. J. Tursina, "Sentiment Analysis of School Zoning System on Youtube Social Media Using the K-Nearest Neighbor with Levenshtein Distance Algorithm," in *2019 7th International Conference on Cyber and IT Service Management, CITSM 2019*, 2019, no. May, pp. 1–4. doi: 10.1109/CITSM47753.2019.8965407.