|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Home** | **About** | **User Home** | **Categories** | **Search** | **Current** | **Archives** | **Announceme** | **nts** |  | |
| **Home** > **User** > **Author** > **Submissions** > **#9569** > **Summary** | | | | | | | |  | **QUICK LINKS** | |
| **SUMMARY**  **REVIEW**  **EDITING**  **Submission**  Authors Anisya Anisya, Azwar Yusuf  Title EKSTRAKSI TEKS METADATA ARTIKEL DENGAN METODE REGEX   |  |  |  | | --- | --- | --- | | Original file | **9569-21457-2-SM.DOCX** 2025-07-29 |  | | Supp. files | None | **ADD A SUPPLEMENTARY FILE** | | Submitter | Mrs Anisya Anisya |  | | Date submitted | July 29, 2025 - 04:18 PM |  | | Section | Articles |  | | Editor | None assigned |  | | **Status**  Status | Awaiting assignment |  | | Initiated | 2025-07-29 |  | | Last modified 2025-07-29 | | |   **Submission Metadata**  **EDIT METADATA**  **AUTHORS**  Name Anisya Anisya  ORCID iD **https://orcid.org/0000-0002-5090-3211**  Affiliation Institut Teknologi Padang  Country Indonesia  Bio Statement —  Principal contact for editorial correspondence.    Name Azwar Yusuf  Affiliation Institut Teknologi Padang  Country Indonesia  Bio Statement —  **TITLE AND ABSTRACT**  Title EKSTRAKSI TEKS METADATA ARTIKEL DENGAN METODE REGEX    Proses pencarian informasi dalam suatu kumpulan artikel pada arsip jurnal Teknoif dilakukan secara manual dengan membuka dokumen satu per satu, yang membutuhkan waktu tambahan. Sistem arsip jurnal Teknoif ini dibuat untuk dapat melakukan pengelompokan dan pengelolaan informasi metadata artikel pada arsip jurnal Teknoif. Pembangunan dilakukan dengan menggunakan metode Extreme Programming (XP) dengan 2 kali iterasi yang berfokus pada pengkodean. Dalam pembangunan sistem ekstraksi metadata artikel untuk penemuan dan pengelompokan metadata artikel  Abstract menggunakan Regular Expression (RegEx) dilakukan dengan tiga tahap, yaitu ekstraksi teks dari file pdf, pembersihan data (cleaning) dengan tiga kriteria, dan ekstraksi metadata mencakup empat kriteria. Sistem ini berhasil mengekstraksi informasi penting seperti judul, nama penulis, afiliasi, dan abstrak dari dokumen pdf secara otomatis. Seluruh proses berjalan sesuai dengan fungsionalitas yang diharapkan dan telah berhasil diuji menggunakan metode black-box testing tanpa ditemukan | | | | | | | |  |  | **Editorial Team** |
| **Reviewer** |
| **Focus & Scope** |
| **Author Guidelines** |
| **Publication Ethics** |
| **Open Access Policy** |
| **Peer Review Proces** |
| **Online Submission** |
| **OPEN ACCESS**  TEMPLATE |





E - ISSN dan P - ISSN



PLAGIARISM TOOL



REFERENCES TOOL



**USER**

You are logged in as...

**anisyanisa**

**My Journals**

**My Profile**

**Log Out**

**AUTHOR**

Submissions

**Active**

(3)

**Archive**

(0)

**New Submission**

**NOTIFICATIONS**

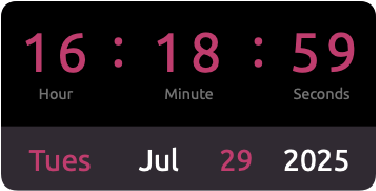
**View**

**Manage**

**JOURNAL CONTENT**

Search

Search Scope



bug yang signifikan. Dengan demikian, sistem ini dapat membantu dalam pengelolaan arsip jurnal

Teknoif .

**INDEXING**

Academic discipline

and sub-disciplines

—

Subject classification

—

Keywords

Metadata Artikel, Regular Expression (RegEx), Extreme Programming (XP)

Language

en

**SUPPORTING AGENCIES**

Agencies

—

**REFERENCES**

References

[1]

F. Reza, I. K. D. Indah, and M. Ropianto, “Perancangan Dan Implementasi Institutional Repository

Dengan Metadata Dublin Core,” komtekinfo, pp. 125–132, Dec. 2022, doi:

10.35134

/komtekinfo.v

9i4.318.

F. Rahutomo, D. A. Irawati, and M. A. E. Pramudita, “Pengembangan Sistem Ekstraksi Metadata

[2]

Artikel ilmiah secara Otomatis,” JTIIK, vol. 6, no. 2, pp. 123–128, Feb. 2019, doi:

10.25126

2019621227.

/jtiik.

Y. Puspitarani, Y. Syukriyah, and U. Widyatama, “Pemanfaatan Optical Character Recognition Dan

[3]

Text Feature Extraction Untuk Membangun Basisdata Pengaduan Tenaga Kerja,” vol. 4, no. 4, 2020.

D. N. Fadhillah and A. Rachman, “IMPLEMENTASI REGEX PADA PEMBERIAN KOMENTAR

[4]

KODE PROGRAM HTML,” JARS, vol. 2, no. 1, pp. 1–5, Dec. 2023, doi: 10.24929/jars.v2i1.3078.

[5]

J. M. Bintang, M. F. Ashshidiq, and H. F. Dzakwan, “Penerapan Algoritma String Matching dan

Regular Expression pada Aplikasi Kamus Besar Bahasa Indonesia (KBBI),” bios, vol. 4, no. 1, pp.

34

4i1.57.

–41, Mar. 2023, doi: 10.37148/bios.v

[6]

N. A. Septiani and L. D. Yanti, “SISTEM INFORMASI PEMASANGAN IKLAN KORAN PADA PT.

HARIAN TOPSKOR DENGAN METODE EXTREME PROGRAMMING (XP),” vol. 06, Dec. 2021.

F. A. T. Tobing and R. Nainggolan, “ANALISIS PERBANDINGAN PENGGUNAAN METODE

[7]

BINARY SEARCH DENGAN REGULAR SEARCH EXPRESSION,” jmika, vol. 4, no. 2, pp. 168–172,

Oct. 2021, doi: 10.46880/jmika.Vol4No2.pp168-172.

[8]

I. Ahmad, R. I. Borman, J. Fakhrurozi, and G. G. Caksana, “Software Development Dengan

Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android,” vol. 5,

no. 2, 2020.

Q. E. Fazrin, T. Lisnawati, S. Nurhayati, J. B. Satya, and D. Alamsyah, “Penerapan Metode

[9]

Pengembangan Sistem Extreme Programing (XP) Pada Aplikasi Presensi Karyawan dengan QR

Code,” bits, vol. 3, no. 3, pp. 164–170, Dec. 2021, doi: 10.47065/bits.v3i3.1018.

R. I. Borman, A. T. Priandika, and A. R. Edison, “Implementasi Metode Pengembangan Sistem

[10]

Extreme Programming (XP) pada Aplikasi Investasi Peternakan,” vol. 8, no. 3, 2020.

K. Nistrina and L. Sahidah, “UNIFIED MODELLING LANGUAGE (UML) UNTUK

[11]

PERANCANGAN SISTEM INFORMASI PENERIMAAN SISWA BARU DI SMK MARGA INSAN

KAMIL,” Jurnal Sistem Informasi, vol. 04, 2022.

B. J. M. Putra, A. Fu’adi, and D. A. F. Yuniarti, “Analisa dan Rancangan Sistem Informasi

[12]

Pariwisata Pacitan dengan UML dan ERD,” vol. 7, no. 1, 2022.

[13]

K. Afiifah, Z. Fira Azzahra, and A. Dwi Anggoro, “Analisis Teknik Entity-Relationship Diagram

dalam Perancangan Database Sebuah Literature Review,” INTECH (Informatika dan Teknologi), vol.

, no. 1, pp. 8–11, Apr. 2022, doi: 10.54895/intech.v

3i1.1261.

3

[14]

F. Azmi, “SISTEM PARKIR DENGAN RADIO FREQUENCY IDENTIFICATION BERBASIS

GELOMBANG RADIO,” vol. 3, 2023.

Ni Nyoman Emang Smrti, A. I Putu Gd Sukenada, D. T. R. Ni Kadek, A. Adnan, and J. Pande

[15]

Putu Ode, “Flowgorithm Sebagai Penunjang Pembelajaran Algoritma dan Pemrograman,”

bangkitindonesia, vol. 12, no. 1, pp. 56–64, Mar. 2023, doi: 10.52771/bangkitindonesia.v12i1.218.

Z. Tuasamu et al., “Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan DFD

[16]

dan Flowchart Pada Bisnis Porobico,” Jurnal Bisnis dan Manajemen (JURBISMAN), vol. 1, no. 2, Art.

no. 2, May 2023, doi: 10.61930/jurbisman.v1i2.181.

**The Journal has been Indexed by :**



**BARCODE**

**P-ISSN**

**dan E-ISSN**



All

Search

Search

Browse

**By Issue**

**By Author**

**By Title**

**By Sections**

**By Identify Types**

**Other Journals**

**Categories**

**INFORMATION**

**For Readers**

**For Authors**

**For Librarians**



**Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI) . Prodi Teknik Komputer, Fakultas Teknik,**

**Universitas Serambi Mekkah**



**JNKTI is licensed under a**

**Creative Commons Attribution-ShareAlike 4.0 International License**

**.**

**EKSTRAKSI TEKS METADATA ARTIKEL DENGAN METODE REGEX**

|  |  |
| --- | --- |
| *Submitted Date :* | *Accepted Date :* |

**Abstrak** - Proses pencarian informasi dalam suatu kumpulan artikel pada arsip jurnal Teknoif dilakukan secara manual dengan membuka dokumen satu per satu, yang membutuhkan waktu tambahan. Sistem arsip jurnal Teknoif ini dibuat untuk dapat melakukan pengelompokan dan pengelolaan informasi metadata artikel pada arsip jurnal Teknoif. Pembangunan dilakukan dengan menggunakan metode Extreme Programming (XP) dengan 2 kali iterasi yang berfokus pada pengkodean. Dalam pembangunan sistem ekstraksi metadata artikel untuk penemuan dan pengelompokan metadata artikel menggunakan Regular Expression (RegEx) dilakukan dengan tiga tahap, yaitu ekstraksi teks dari file pdf, pembersihan data (cleaning) dengan tiga kriteria, dan ekstraksi metadata mencakup empat kriteria. Sistem ini berhasil mengekstraksi informasi penting seperti judul, nama penulis, afiliasi, dan abstrak dari dokumen pdf secara otomatis. Seluruh proses berjalan sesuai dengan fungsionalitas yang diharapkan dan telah berhasil diuji menggunakan metode black-box testing tanpa ditemukan bug yang signifikan. Dengan demikian, sistem ini dapat membantu dalam pengelolaan arsip jurnal Teknoif .

*Kata Kunci : Metadata Artikel, Regular Expression (RegEx), Extreme Programming (XP).*

***Abstract*** *- The process of searching for information in a collection of articles in the Teknoif journal archive is done manually by opening documents one by one, which requires additional time. This Teknoif journal archive system was created to be able to group and manage article metadata information in the Teknoif journal archive. Development was carried out using the Extreme Programming (XP) method with 2 iterations that focused on coding. In developing an article metadata extraction system for finding and grouping article metadata using Regular Expression (RegEx), it was carried out in three stages, namely text extraction from PDF files, data cleaning with three criteria, and metadata extraction covering four criteria. This system successfully extracted important information such as title, author name, affiliation, and abstract from PDF documents automatically. The entire process ran according to the expected functionality and was successfully tested using the black-box testing method without any significant bugs found. Thus, this system can assist in the management of the Teknoif journal archive.*

*Keywords : Article Metadata, Regular Expression (RegEx), Extreme Programming (XP).*

1. Pendahuluan

Proses pendataan arsip jurnal *teknoif* dilakukan secara manual dengan melakukan proses pencarian informasi dalam kumpulan artikel dengan membuka dokumen satu per satu, yang membutuhkan waktu tambahan. *Metadata* merupakan elemen untuk menjabarkan suatu data secara detail berupa informasi terstruktur yang dapat mendeskripsikan, menjelaskan dan mempermudah pencarian sehingga lebih mudah untuk pengambilan kembali, penggunaan dan pengelolaan suatu informasi[1]. *Metadata* artikel berupa judul, penulis, instansi, surel penulis, abstrak, kata kunci, dan daftar pustaka[2]. Saat pengelola arsip jurnal *teknoif* tidak menerapkan sistem *metadata*, maka proses pengelompokan informasi dengan standar kebutuhan untuk arsip jurnal *teknoif* yang berisi judul, penulis, afiliasi dan abstrak kurang optimal.

Ekstraksi teks sendiri memiliki peran dalam proses *metadata*, *extraction* atau lebih umum dikenal dengan *Information Extraction*, akan menentukan bagian-bagian dari teks berupa informasi yang lebih terstruktur [3]. *Regular Expression (ReGex)* mengenali pola yang dapat digunakan untuk mendeskripsikan pola dari kata yang ingin dicari[4]. Dengan penggunaan *ReGex* dapat membuat pola yang kompleks untuk mencari kata-kata dengan karakteristik tertentu[5].

Penggunaan *Regular Expression* (*ReGex*) untuk ekstraksi *Metadata* artikel merupakan metode yang dapat digunakan dalam menemukan informasi *Metadata*, terutama dalam mengidentifikasi judul, nama penulis, afiliasi, dan abstrak berdasarkan pola dari *metadata* artikel. Dengan menggunakan sistem ekstraksi *metadata* diharapkan dapat menjadi lebih cepat dan efisien dibandingkan dengan proses pengamatan satu persatu dari setiap artikel.

*Extreme Programming* merupakan sebuah pendekatan atau model pengembangan perangkat lunak yang mencoba menyederhanakan berbagai tahapan dalam proses pengembangan tersebut sehingga menjadi lebih adaptif dan fleksibel[6]. *Extreme Programming* tepat digunakan dalam perancangan dan pengembangan sistem, yang membutuhkan sifat fleksibel dan interaksi dengan pengguna untuk menyelesaikan masalah yang ada sehingga dapat membantu dalam mengelola arsip jurnal *teknoif*.

1. METODE

Penelitian yang bertujuan untuk meneliti proses ekstraksi *teks metadata* artikel menggunakan metode *Regular Expression* (*RegEx*) pada sistem pengelompokan dan pengelolaan informasi *metadata* artikel. Sistem ini dibuat untuk dapat melakukan mengumpulkan informasi *metadata* artikel*.*

Metode Pengumpulan Data

Informasi dan data yang dikumpulkan dilakukan dengan metode sebagai berikut:

* 1. Studi Literatur

Melakukan pengumpulan data dengan membaca beberapa literatur yang mendukung dan berkaitan dengan penelitian, sehingga dapat memberikan informasi yang relevan dan mendalam. Literatur ini membantu dalam memahami teori, metode, dan temuan terdahulu sebagai dasar penyusunan penelitian.

* 1. Wawancara

Wawancara dilakukan untuk mendapatkan informasi maupun data – data yang dibutuhkan dengan melakukan wawancara dengan pihak yang terkait sebagai pengelola jurnal. Pengelola jurnal mengatakan bahwa permasalahan yang dialaminya sekarang perlu melakukan peninjauan secara satu tersatu untuk menemukan informasi *metadata* artikel dalam mengelola jurnal. Proses meninjau satu persatu informasi dimulai dari judul, penulis, afiliasi dan abstrak kemudian di cocokkan antara penulis satu hingga seterusnya jika penulis lebih dari satu kemudian dihubungkan ke judul artikel.

* 1. *Dataset Metadata* Artikel

*Dataset metadata* artikel meliputi sekumpulan data yang memuat informasi penting mengenai artikel ilmiah, yang berfungsi untuk menjelaskan dan memberikan konteks terhadap isi artikel tersebut. *Metadata* artikel yang digunakan untuk pengelolaan serta pencarian data.



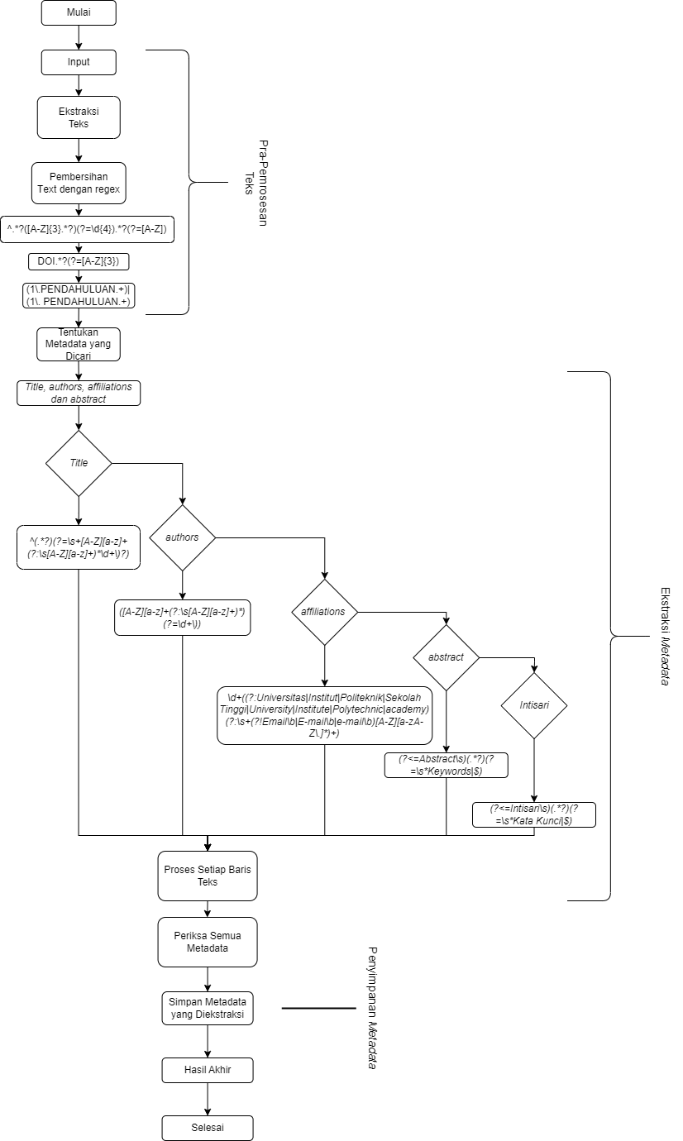
Gambar 1. *Dataset* *metadata* artikel

Pada Gambar 1 merupakan *dataset metadata* artikel meliputi:

1. Judul Artikel yang berisikan gambaran singkat isi artikel.
2. Penulis yang berisikan nama orang atau kelompok yang menulis artikel.
3. Afiliasi yang berisikan institusi atau organisasi asal penulis.
4. Abstrak yang berisikan ringkasan artikel di antaranya tujuan, metode, hasil, dan kesimpulan.

Metode Ekstraksi *Metadata*

Teknik ekstraksi teks dari dokumen yang diunggah untuk mendapatkan *metadata* artikel. *Metadata* untuk artikel adalah informasi mengenai artikel. Misalnya data pengarang, judul artikel, tahun, tanggal publikasi. Dapat pula merupakan bahan deskriptif seperti kata kunci dan abstrak. Sehingga, untuk memudahkan pengarsipan dan pencarian kembali sebuah artikel ilmiah dibutuhkan sebuah *metadata* dari artikel ilmiah itu sendiri [2]. Tujuan ekstraksi *metadata* adalah untuk mengidentifikasi informasi penting dalam artikel tanpa harus membuka dokumen satu per satu. Berikut badan alur dari proses ekstasi *metadata*:



Gambar 2. Ekstraksi *Metadata*

Pada Gambar 2 merupakan bagan alur proses untuk mencapai tujuan tersebut, penggunaan *Regular Expression (RegEx)* sebagai teknik utama ekstraksi *metadata*. Berikut tahapan utama yang dilakukan dalam proses ekstraksi teks menggunakan *RegEx*:

1. Pra-Pemrosesan Teks

Pra-pemrosesan bertujuan mempersiapkan teks dari dokumen agar dapat diproses oleh model *RegEx*. Alur *pra-pemrosesan* teks agar teks dapat di proses oleh sistem yang meliputi di antaranya:

1. Ekstraksi teks dari dokumen PDF

Agar teks dari sebuah PDF dapat di baca sistem perlu menggunakan ekstraksi teks. Dalam study kasus ini melakukan pembatasan hanya melakukan ekstraksi pada halaman pertama PDF.

1. Pembersihan teks

Hapus informasi yang tidak digunakan seperti *header* dan *footer*.

Contoh:

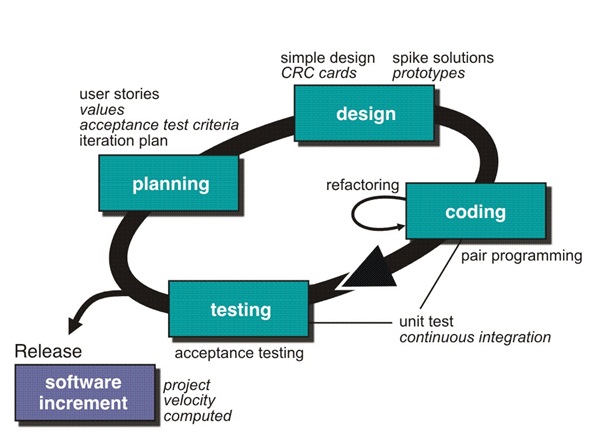
1. *Header* : “Haris Suryamen, et.al /Jurnal Teknoif Teknik Informatika Institut Teknologi Padang - Vol. 12 No. 2 (2024) 70-76”.
2. *Footer*: “DOI : https://doi.org/10.21063/jtif.2024.V12.12.70-76 Creative Commons Attribution-Share Alike 4.0 International License (CC BY-SA 4.0)”.
3. Ekstraksi *Metadata*

Ekstraksi *metadata* menggunakan *Regular Expression (Regex)* yang merupakan metode atau algoritma pencarian teks yang memiliki konsep dasar pencarian bukan hanya kepada teks secara spesifik, namun melakukan pencarian berdasarkan pola dari teks tersebut. konsep pencarian ini dapat melakukan pencarian tanpa perlu mengetahui secara lengkap dan tepat data yang akan dicari[7]. *Regex* digunakan untuk mengekstrak informasi spesifik berdasarkan ciri-ciri atau pola seperti:

1. Judul: Terletak di bagian pertama pada artikel.
2. Penulis: Cari pola yang umum digunakan untuk menyebutkan penulis yang tertulis setelah penulisan judul.
3. Afiliasi: Identifikasi afiliasi penulis yang biasanya berada setelah nama penulis.
4. Abstrak: Temukan bagian yang biasanya berlabel "Abstrak" dan “*Intisari*” kemudian ambil teks yang mengikuti label tersebut.
5. Penyimpanan *Metadata*

Penyimpanan berhasil diekstraksi yang telah dikelompokkan berdasarkan kategori (judul, penulis, afiliasi, abstrak) dan disimpan dalam *database* seperti *MySQL* untuk mempermudah pengelolaan.

Metode *Extreme* *Programming*



Gambar 3. Tahapan *Extreme Programming* (XP)[8]

*Metode Extreme Programming* (XP) merupakan metodologi pengembangan *agile software development methodologies* yang memiliki fokus pada *pengkodean* (*coding*) yang merupakan aktivitas utama dalam semua tahapan pada siklus pengembangan perangkat lunak[9]. Metode *Extreme Programming (XP)* dipilih karena memiliki tingkat responsif yang baik terhadap perubahan. Tahapan-tahapan dalam metode *Extreme Programming* (XP) meliputi *planning* (perencanaan), *design* (perancangan), *coding* (pengkodean), dan testing (pengujian) [10].

1. Planning

Pada tahap *Planning* dilakukan wawancara dengan pihak pengelola jurnal *teknoif* mengenai alur proses *input* data publikasi pada sistem arsip jurnal *teknoif* yang terjadi saat ini. Berdasarkan wawancara yang telah di lakukan, berikut hasil wawancara :

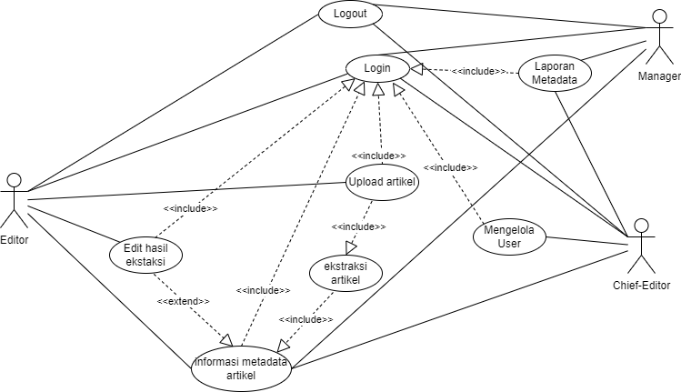
* 1. Proses pengelolaan *metadata* artikel dilakukan dengan membuka setiap artikel untuk menemukan informasi *metadata.*
  2. Memasukkan informasi *metadata* tersebut satu per satu, yang dimulai dari judul,penulis, afiliasi dan abstrak.
  3. Melakukan pencocokan satu per satu antara penulis pertama hingga seterusnya dengan judul artikel.
  4. Melakukan pencocokan satu per satu antara penulis pertama hingga seterusnya dengan afiliasi.

1. Design

Tahap *design* yang berfokus pada kegiatan merancang aplikasi secara sistematis dan terstruktur. Pada tahap ini, berbagai alat bantu digunakan untuk memvisualisasikan sistem yang akan dibangun. Beberapa alat yang umum digunakan dalam tahap desain antara lain: *use case diagram*, *class diagram*, dan *activity diagram*[10].

1. *Use Case Diagram*

*Use Case Diagram* memiliki fungsi untuk menjelaskan interaksi antara satu atau lebih aktor dengan sistem informasi yang sedang dikembangkan, serta dapat mengilustrasikan berbagai fungsi yang terdapat dalam sistem informasi tersebut[11].

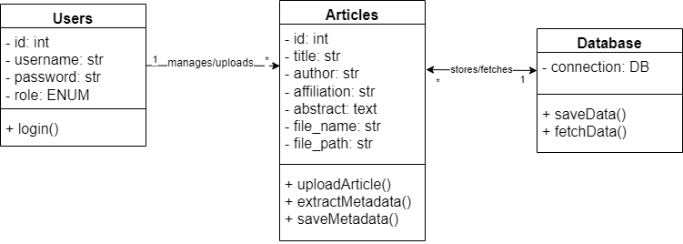


Gambar 4. *Use* *Case* *Diagram*

Pada Gambar 4 menunjukkan interaksi antara tiga lever pengguna, yaitu *Editor, Manager*, dan *Chief-Editor*, dengan fungsi-fungsi yang ada dalam sistem. Adapun peran dari masing-masing level pengguna sebagai berikut:

1. *Editor* memiliki akses untuk mengunggah artikel, melakukan ekstraksi artikel, serta mengedit hasil ekstraksi *metadata* serta dapat melihat informasi *metadata* artikel yang telah berhasil diekstraksi.
2. *Manager* memiliki akses untuk melihat laporan *metadata* sebagai bagian dari tugas pemantauan dan evaluasi.
3. *Chief-Editor* sebagai level tertinggi yang memiliki akses untuk mengelola pengguna serta mengakses seluruh fitur lainnya.
4. *Class Diagram*

*Class Diagram* merupakan representasi struktur logika sistem, mencakup *class-class* dan elemen-elemen yang membentuk model. *Class Diagram* merincikan informasi terkait *class-class* yang ada, atribut yang dimiliki, dan *behavior* (tingkah laku) [12].

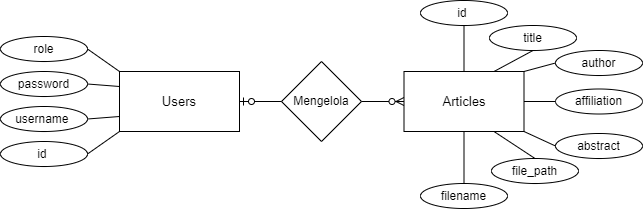


Gambar 5. *Class Diagram*

Pada Gambar 5 menunjukkan struktur *class* utama dalam sistem, yaitu *Users*, *Articles*, dan *Database*. Pada *Class Users* menyimpan informasi pengguna di antaranya *id*, *username*, *password*, dan *role*. Pada *class* ini memiliki fungsi *login()* untuk proses pengecekan pengguna. *Class Articles* digunakan untuk menyimpan *metadata* artikel, yang mencakup atribut seperti *title*, *author*, *affiliation*, *abstract*, *file\_name*, dan *file\_path*. *Class* ini juga memiliki metode *uploadArticle()*, *extractMetadata()*, dan *saveMetadata()* untuk mendukung proses pengunggahan, ekstraksi, dan penyimpanan *metadata* artikel. *Class Database* berfungsi untuk menyimpan dan mengambil data dari *database*. *Class* ini memiliki atribut *connection* untuk menghubungkan ke *database*, serta metode *saveData()* dan *fetchData()* yang berfungsi untuk penyimpanan dan pengambilan data.

1. *Entity Relationship Diagram*

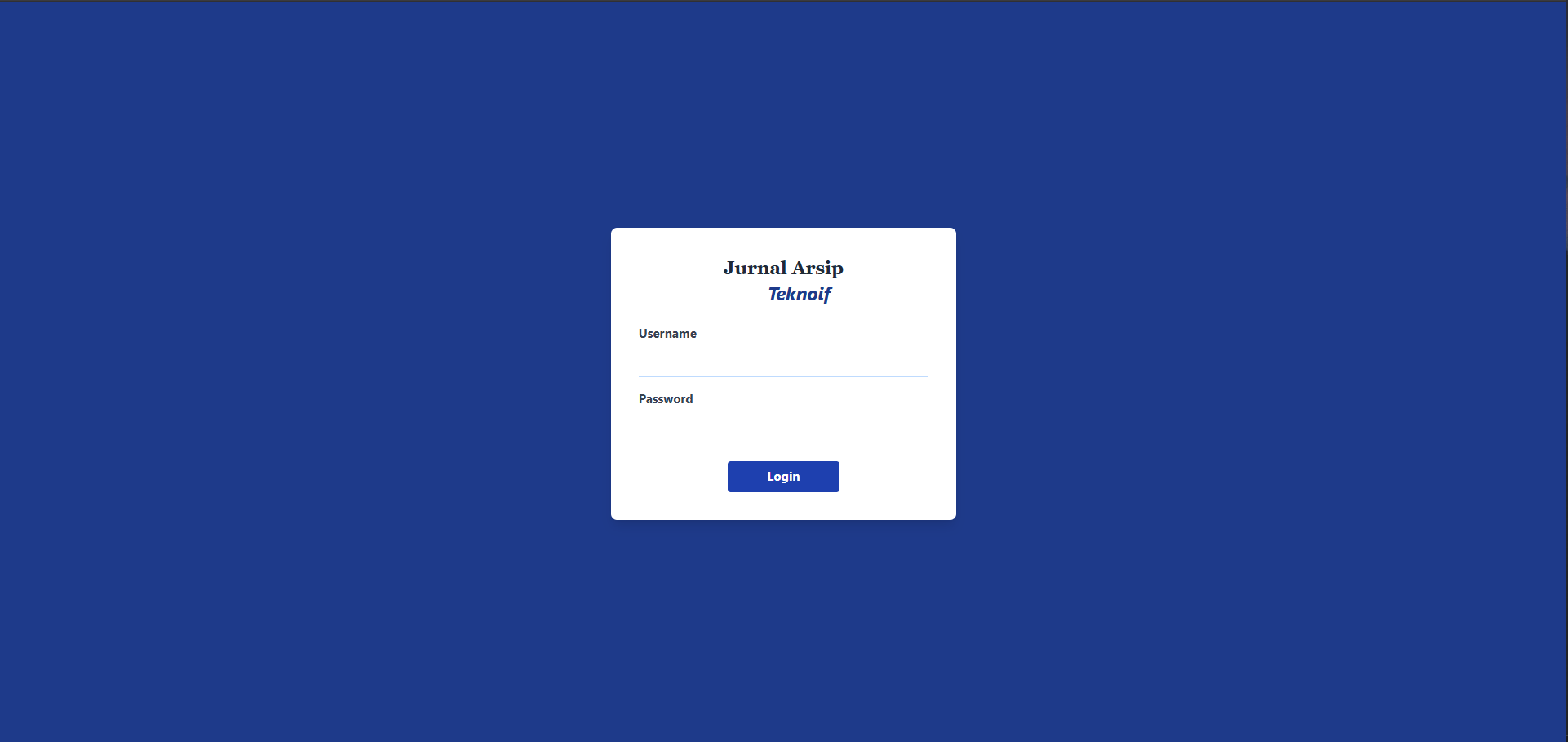
*Entity Relationship Diagram* (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan *database* yang menghubungkan antara data satu dengan yang lain. Fungsi ERD adalah sebagai alat bantu dalam pembuatan *database* dan memberikan gambaran bagaimana kerja *database* yang akan dibuat[13]



Gambar 6. *Entity Relationship Diagram*

Pada Gambar 6 merupakan *entity relationship diagram* memberikan gambaran bagaimana kerangka kerja *database* yang akan dibuat. Yang terdapat entitas *users* dan *articles* dengan atribut-atributnya yang saling terhubung dengan nama relasi mengelola.

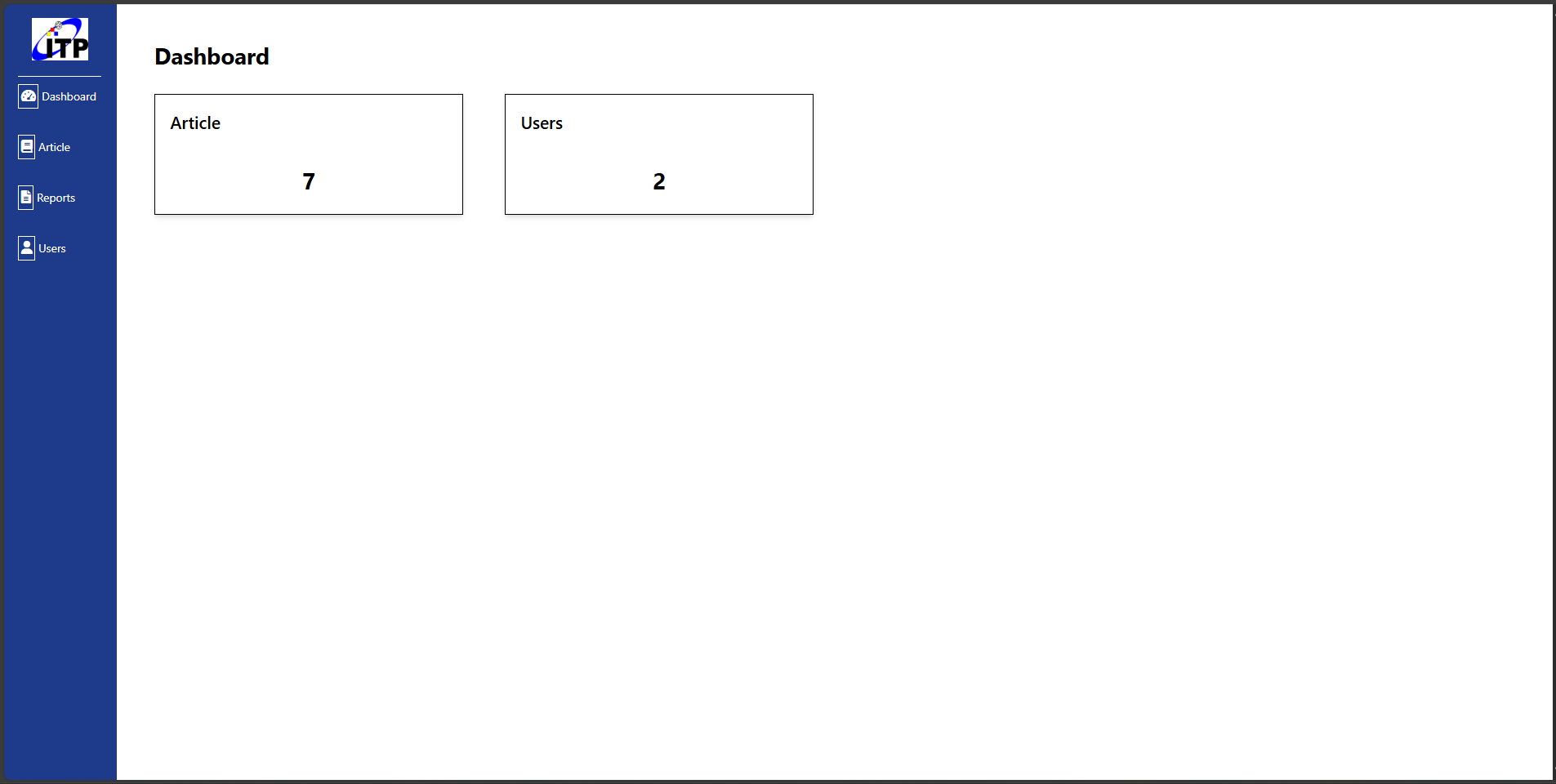
1. HASIL DAN PEMBAHASAN
2. Implementasi *Sistem*
   * 1. Iterasi Pertama
   1. Halaman *Login*



Gambar 7. Halaman *Form Login*

Gambar 7 merupakan tampilan dari halaman *login*. Pada halaman *login*, pengguna diminta untuk mengisi *username* dan *password* yang telah terdaftar untuk melanjutkan di halaman utama pada sistem.

* 1. Halaman *Chief-Editor*
  2. Halaman *Dashboard*



Gambar 8. Halaman *Dashboard Chief-Editor*

Pada Gambar 8 merupakan tampilan dari halaman *dashboard Chief-Editor*, pada halaman *dashboard* terdapat *sidebar navigation* yang dengan pilihan menu *Dashboard*, *Article*, *Reports* dan *User*. Pada menu *Dashboard*, *Chief-Editor* hanyaterdapat melihat informasi dari total artikel serta *user* yang telah terdaftar.

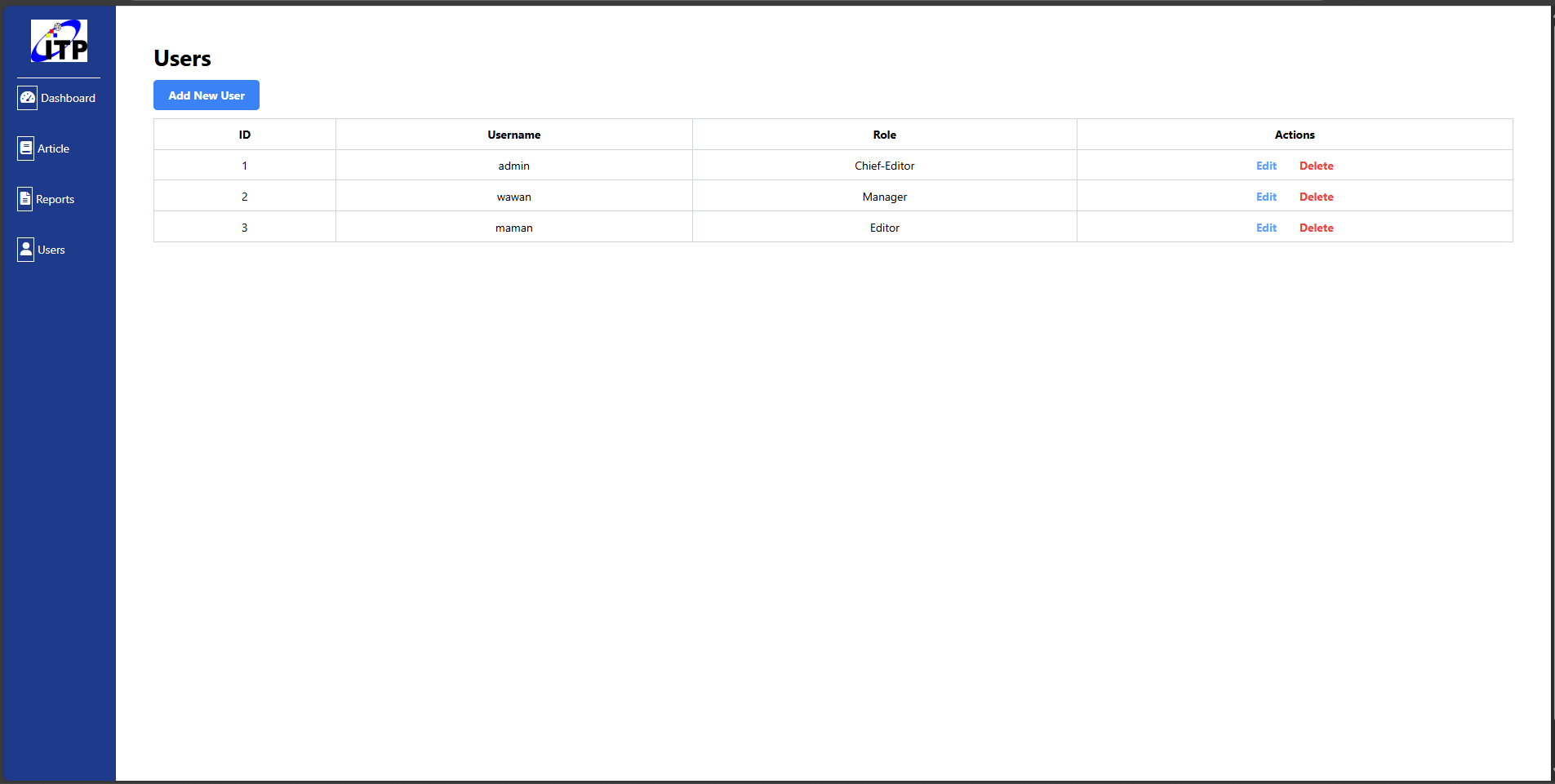
* 1. Halaman *Article*



Gambar 9. Halaman *Article Chief-Editor*

Pada Gambar 9 merupakan tampilan dari halaman *article Chief-Editor*, pada halaman *Article Chief-Editor* terdapat informasi *metadata* artikel dan tombol untuk melakukan *download file* dari artikel yang di proses. Informasi *metadata* artikel diantarnya *title*, *affiliation*, *author* dan *abstract* yang di bungkus dalam *popup*.

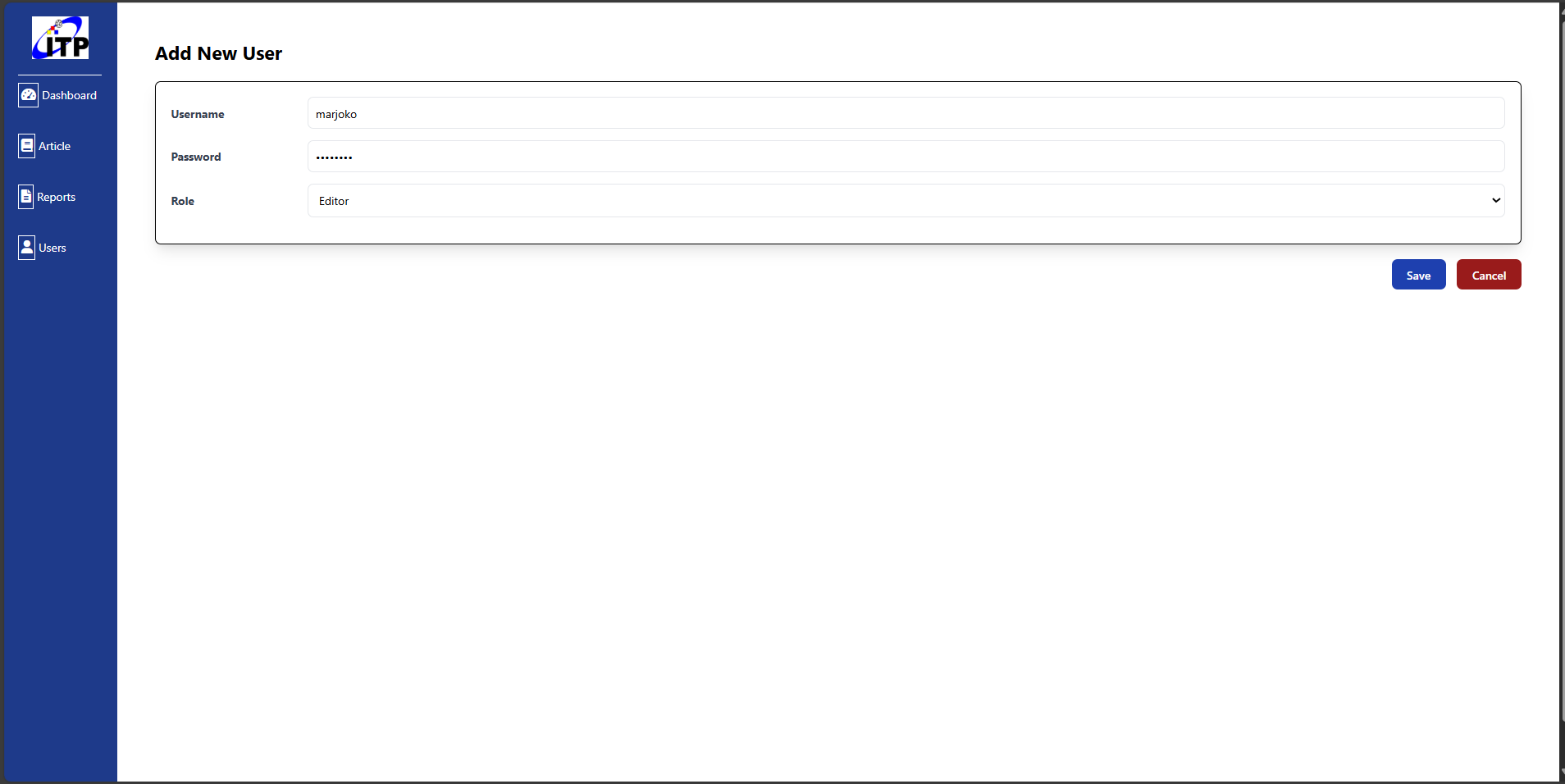
* 1. Halaman *Users*



Gambar 10. Halaman *Users*

Pada Gambar 10 merupakan tampilan dari halaman kelola *user*, pada halaman kelola *user* terlihat informasi data *user* atau pengguna dari sistem arsip *teknoif* di antaranya *username* dan *role*.

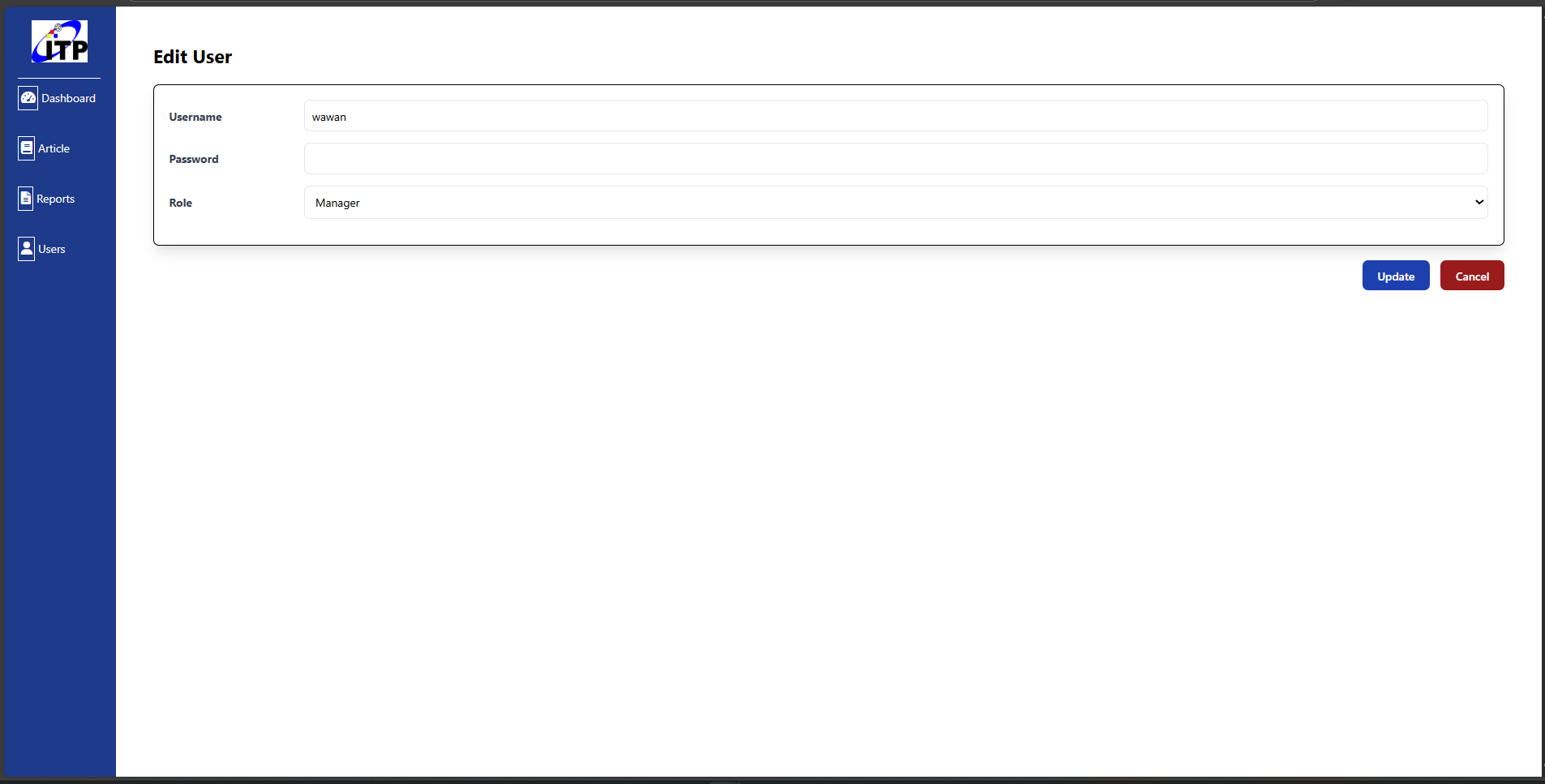
* 1. Halaman *Create Users*



Gambar 11. Halaman *Create Users*

Pada Gambar 11 merupakan tampilan halaman dari *create* *user*, halaman *create* *user* digunakan *chief-Editor* untuk menambahkan pengguna atau *user* baru ke dalam sistem yang meliputi informasi *username, password* dan *role*.

* 1. Halaman *Update Users*

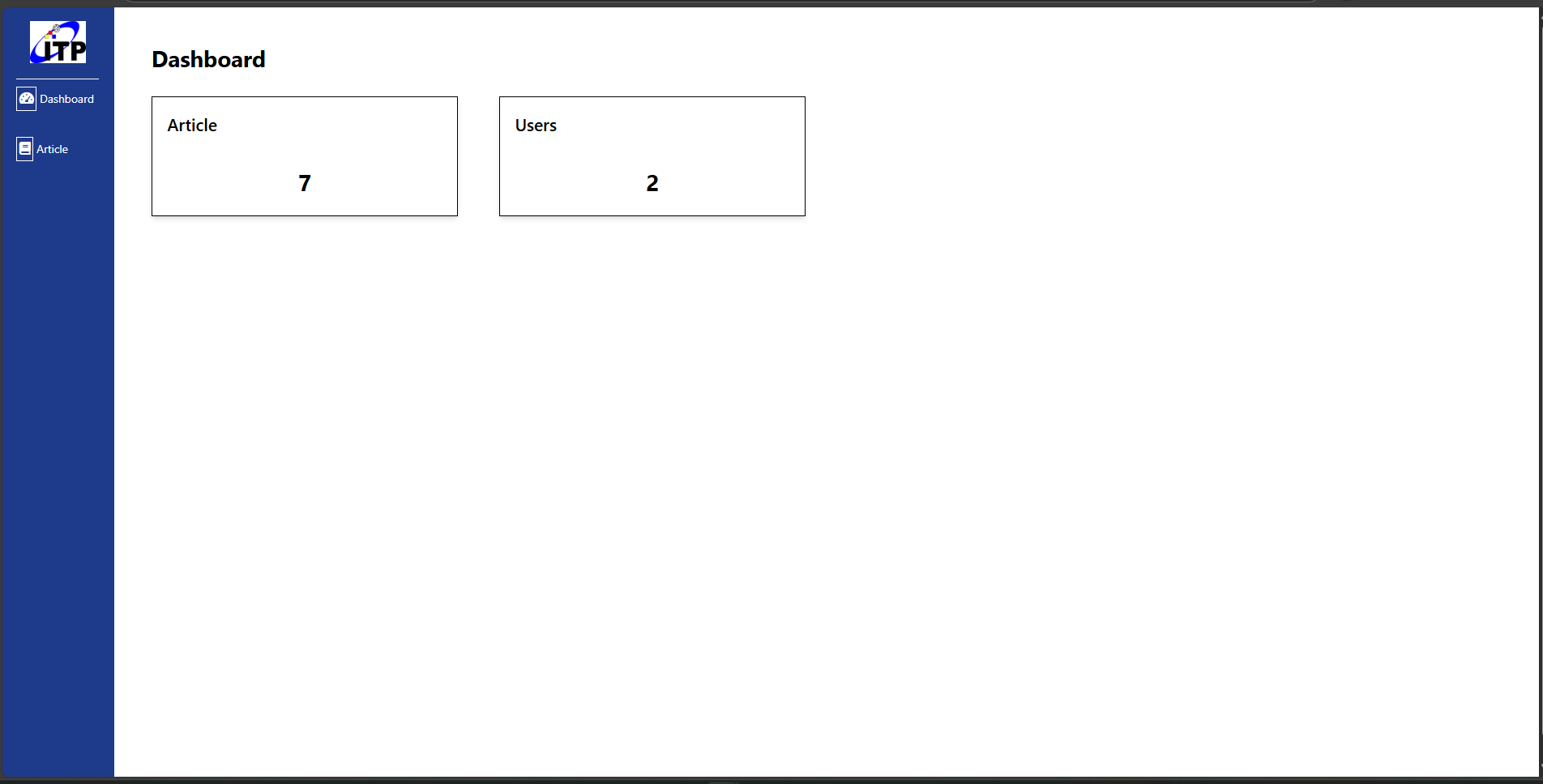


Gambar 12. Halaman *Update Users*

Pada Gambar 12 merupakan tampilan dari halaman *update user*, pada halaman *update user* dapat melakukan perubahan informasi dari akun pengguna di antaranya *username, password* dan *role* dariakun pengguna tersebut.

* 1. Halaman Editor

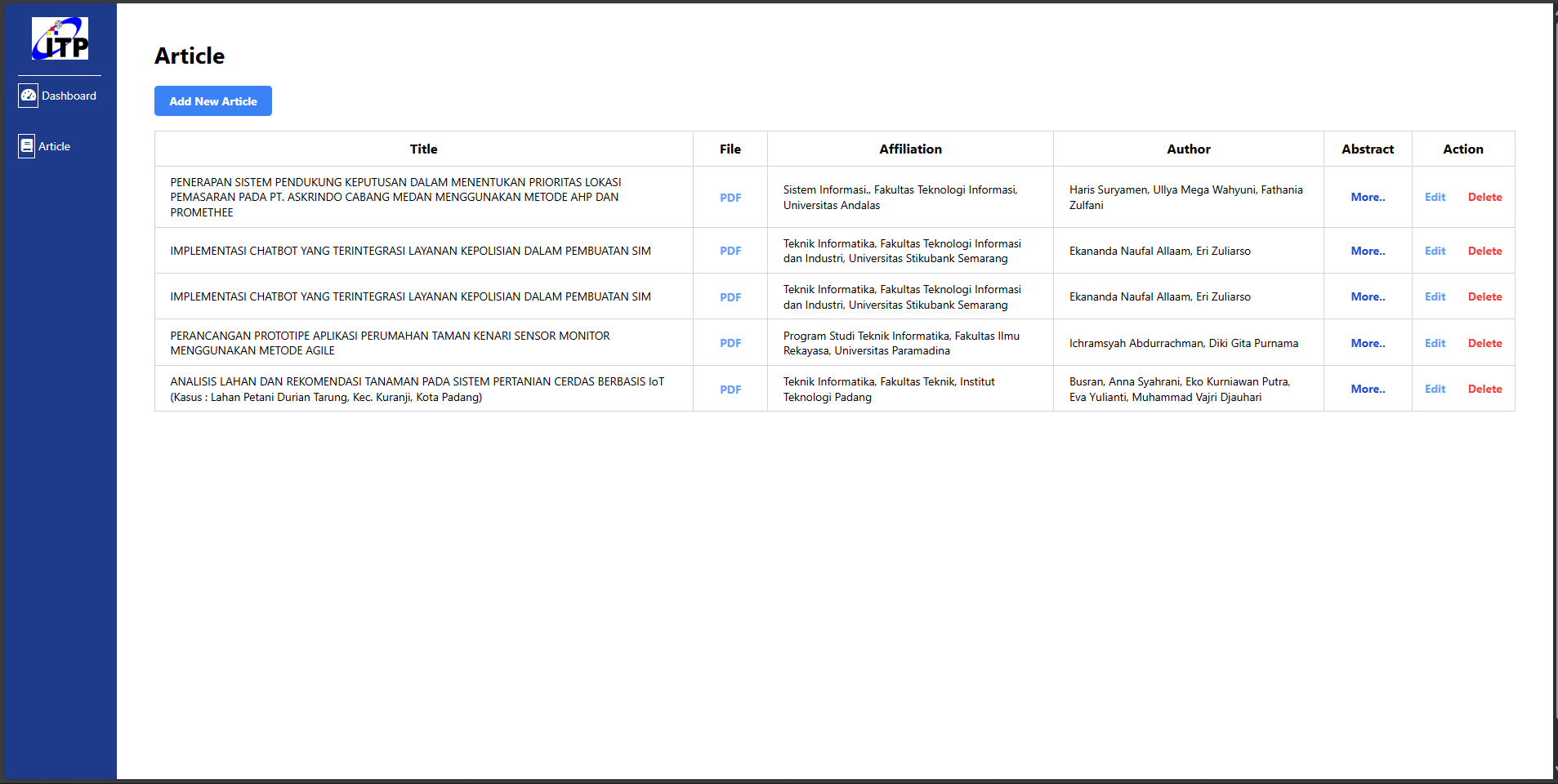
1. Halaman *Dashboard*



Gambar 13. Halaman *Dashboard Editor*

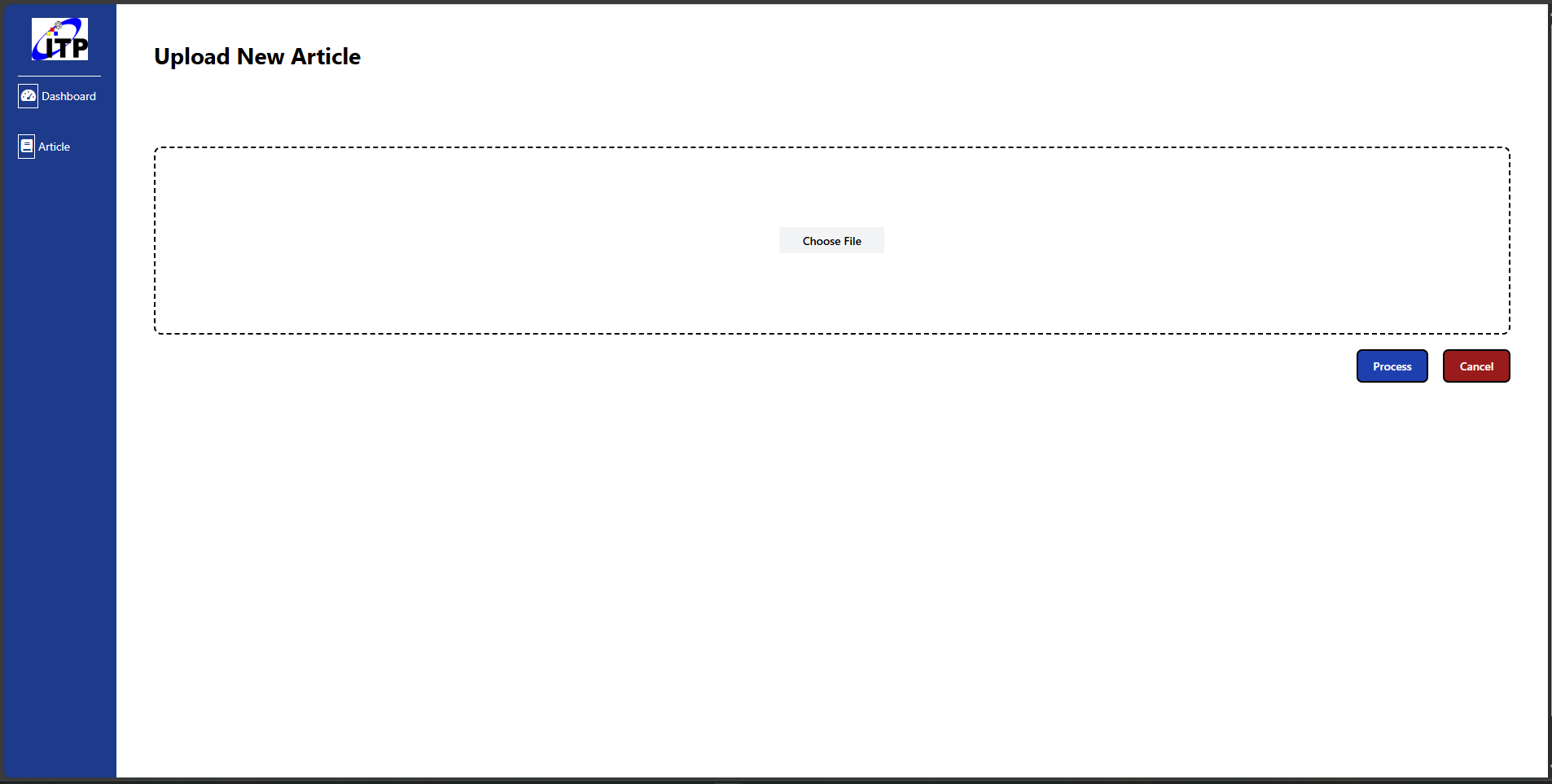
Pada Gambar 13 merupakan tampilan dari halaman *dashboard editor*, pada halaman *dashboard* terdapat *sidebar navigation* yang dengan pilihan menu *Dashboard* dan *Article*. Pada menu *Dashboard*, *editor* hanyaterdapat melihat informasi dari total artikel serta *user* yang telah terdaftar.

1. Halaman *Article*



Gambar 14. Halaman *Article Editor*

Pada Gambar 14 merupakan tampilan dari halaman *article editor*, pada halaman *Article editor* terdapat informasi *metadata* artikel dan tombol untuk melakukan *download file* dari artikel yang di proses. Informasi *metadata* artikel diantarnya *title*, *affiliation*, *author* dan *abstract* yang di bungkus dalam *popup*.



Gambar 15. Halaman *Upload Article Editor*

Pada Gambar 15 merupakan tampilan dari *form upload article* yang akan di proses untuk mendapatkan informasi *metadata* *article,* dengan kebutuhan dari *teknoif* di antaranya judul, nama penulis, afiliasi dan abstrak.



Gambar 16. Halaman *Results* Ekstraksi

Pada Gambar 16 merupakan tampilan dari hasil proses ekstraksi *metadata* artikel. Hasil dari proses ekstraksi dapat di simpan langsung ke *database* dan jika hasil terdapat kesalahan atau kurang sesuai dengan data yang di proses dapat menuju ke proses edit terlebih dahulu.

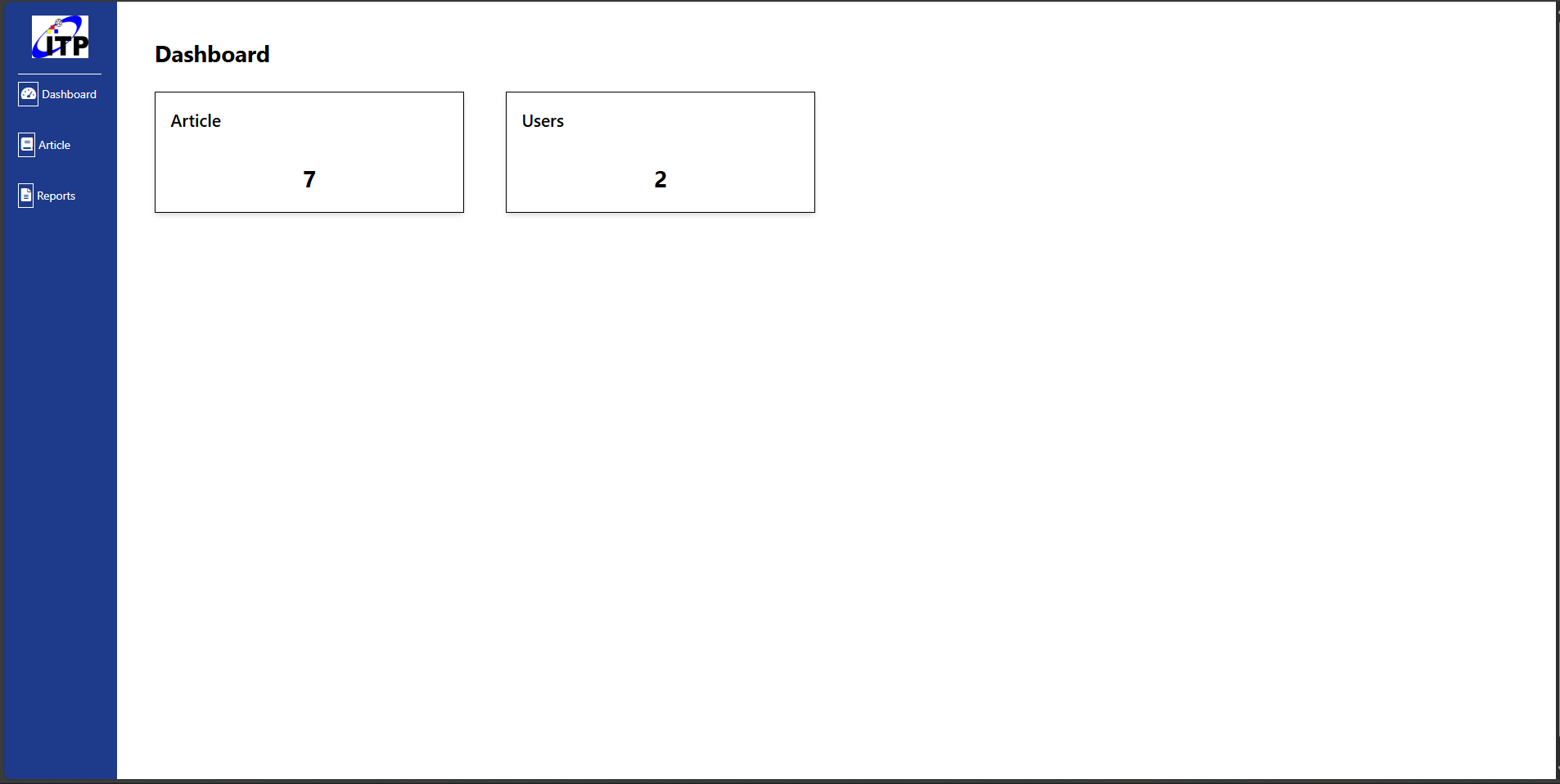


Gambar 17. Halaman *Edit* *Results*

Pada Gambar 17 merupakan tampilan proses edit yang di lakukan untuk memperbaiki hasil ekstraksi sebelum melakukan penyimpanan ke dalam *database*.

* 1. Halaman Manager

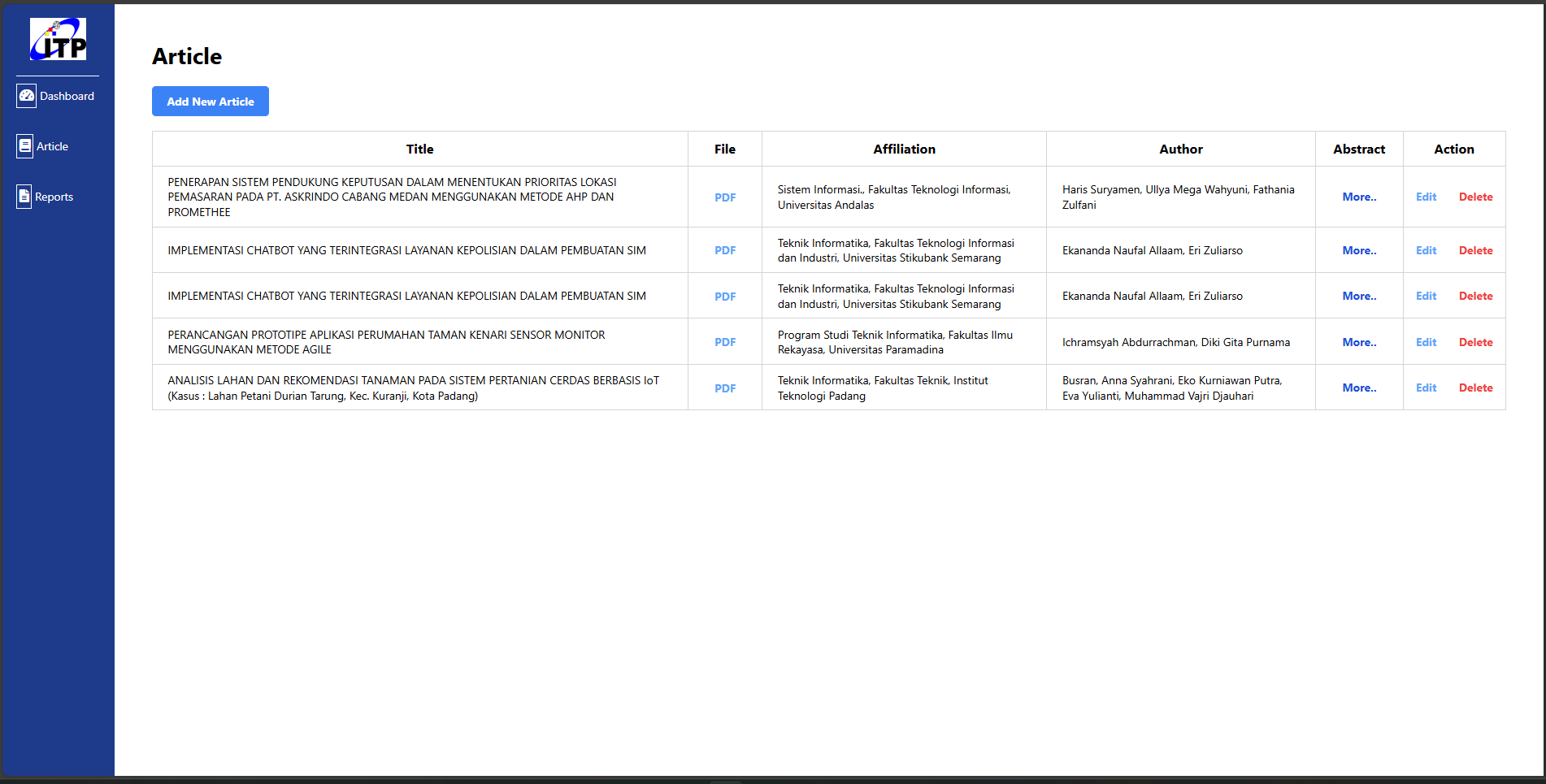
1. Halaman *Dashboard*



Gambar 18. Halaman *Dashboard Manager*

Pada Gambar 18 merupakan tampilan dari halaman *dashboard manager*, pada halaman *dashboard* terdapat *sidebar navigation* yang dengan pilihan menu *Dashboard*, *Article dan reports*. Pada menu *Dashboard*, *manager* dapat melihat informasi dari total artikel serta *user* yang telah terdaftar.

1. Halaman *Article*



Gambar 19. Halaman *Article Manager*

Pada Gambar 19 merupakan tampilan dari halaman *article manager*, pada halaman *article manager* terdapat informasi *metadata* artikel dan tombol untuk melakukan *download file* dari artikel yang di proses. Informasi *metadata* artikel diantarnya *title*, *affiliation*, *author* dan *abstract* yang di bungkus dalam *popup*.

* + 1. IterasiKedua

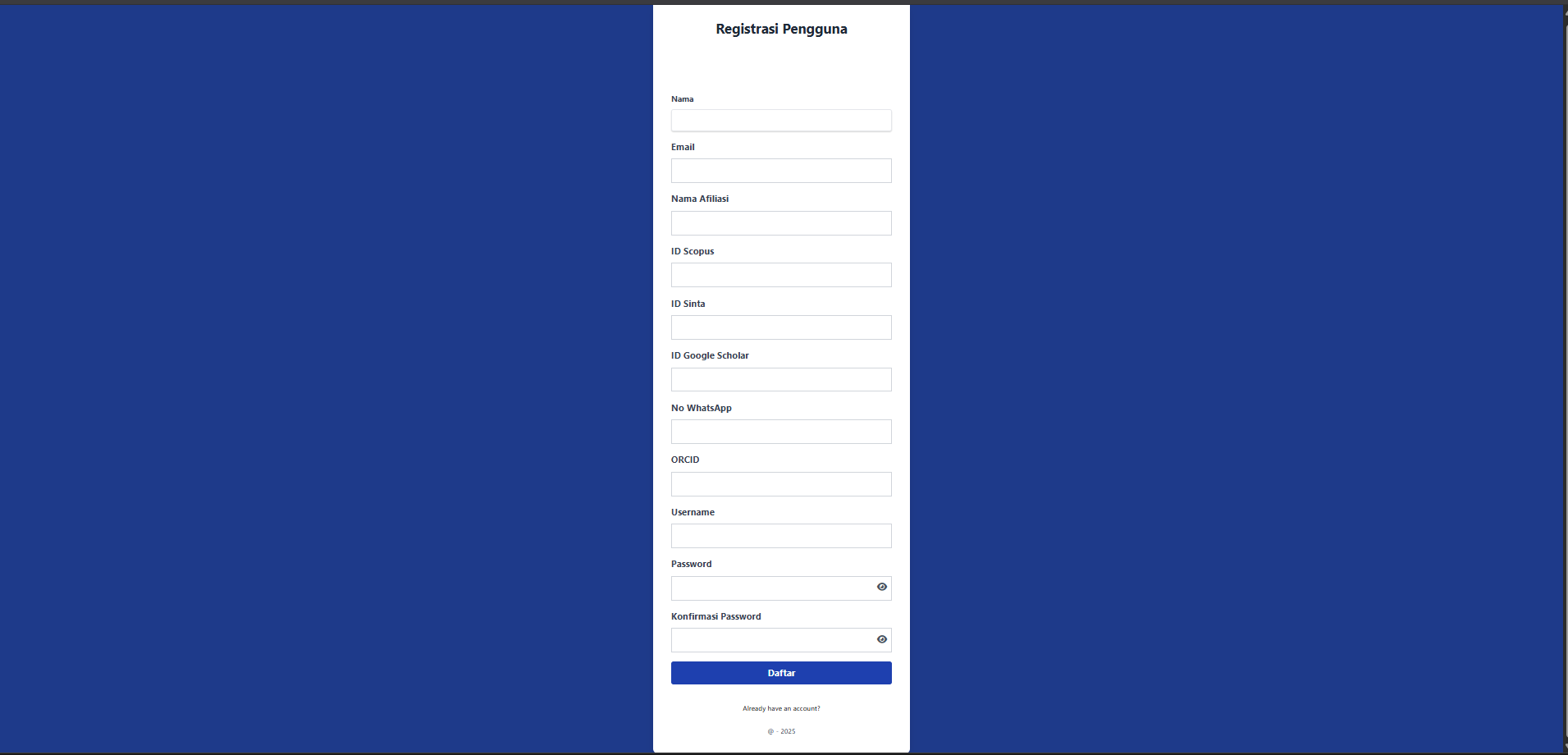
1. Halaman *Login*



Gambar 20 *Form Login*

Pada Gambar 20 merupakan tampilan dari hasil perbaikan pada halaman *login* dengan perbaikan tampilan dan memberikan fitur registrasi atau mendaftarkan diri untuk dapat mengakses sistem.

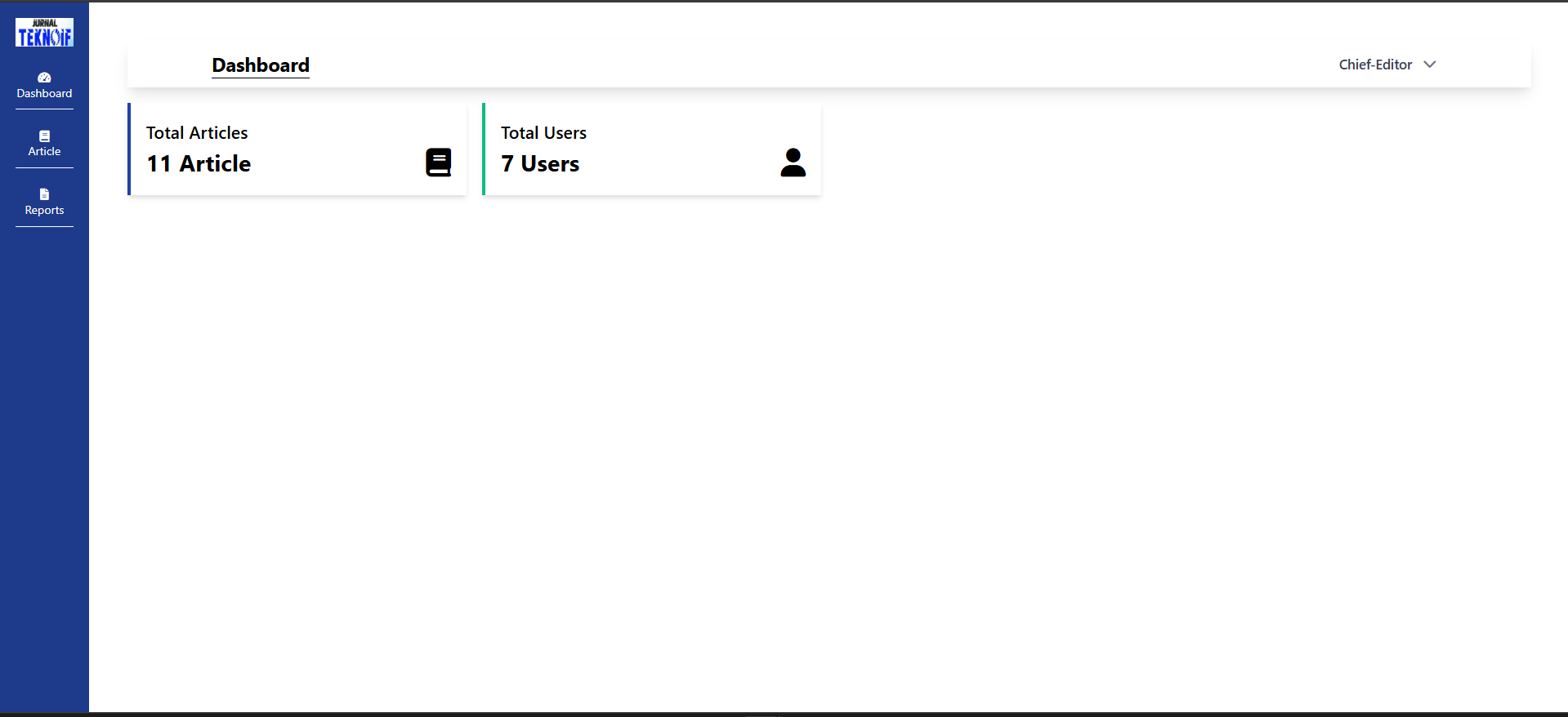
1. Halaman Registrasi



Gambar 21 *Form* Registrasi

Pada Gambar 21 menampilkan halaman registrasi dengan data yang di masukan di antaranya Nama, Email, Nama Afiliasi, *ID\_Scopus, ID\_Sinta, ID\_Google Scholar, No. Whatsapp, ORCIT, Username, Password* serta Konfirmasi Password.

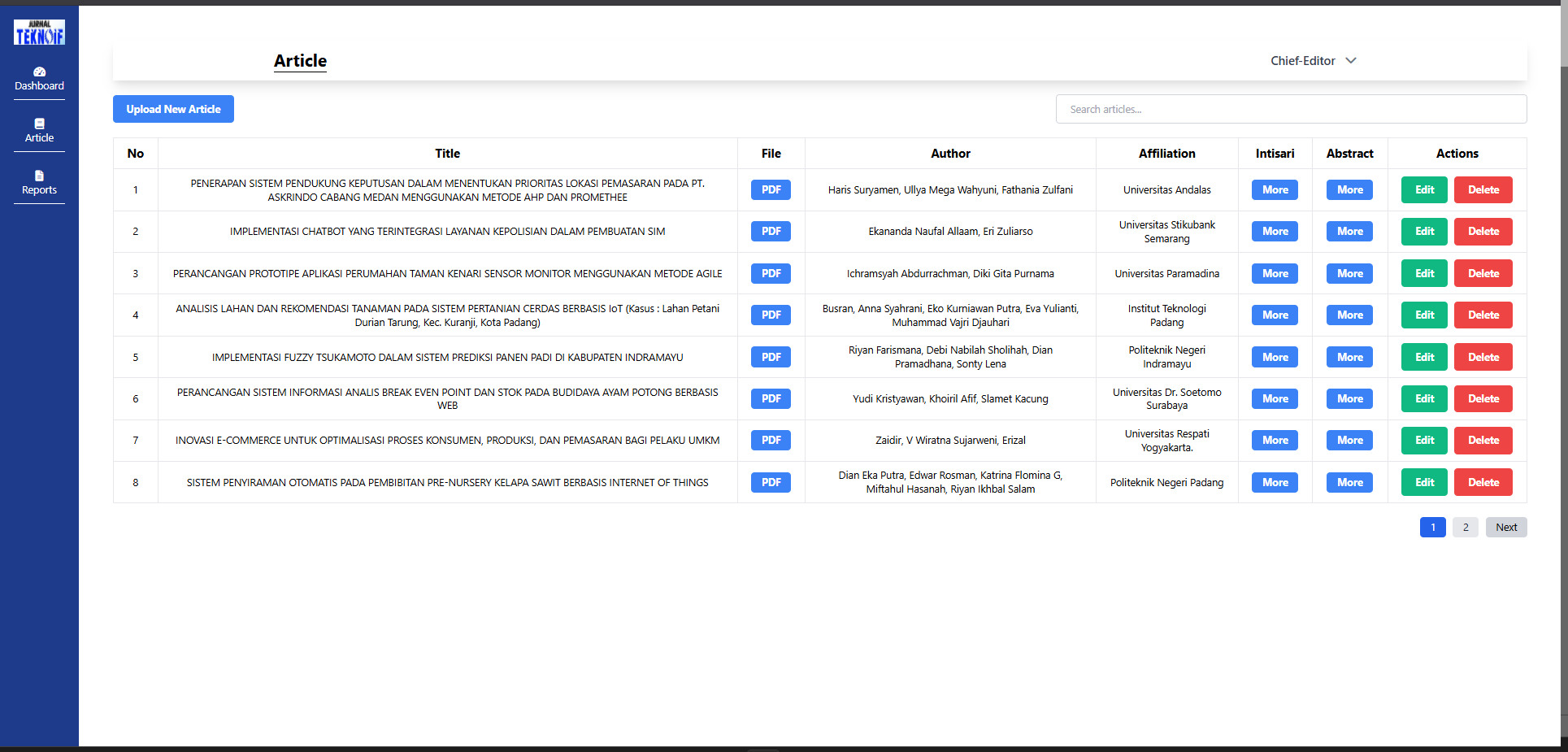
1. Halaman *Chief-Editor*
   * + 1. Halaman *Dashboard*



Gambar 22 *Dashboard Chief-Editor*

Pada Gambar 22 merupakan tampilan hasil dari perbaikan halaman *dashboad chief-editor* dengan perbaikan pada tampilan antar muka dengan penambahan informasi *role* yang sedang aktif.

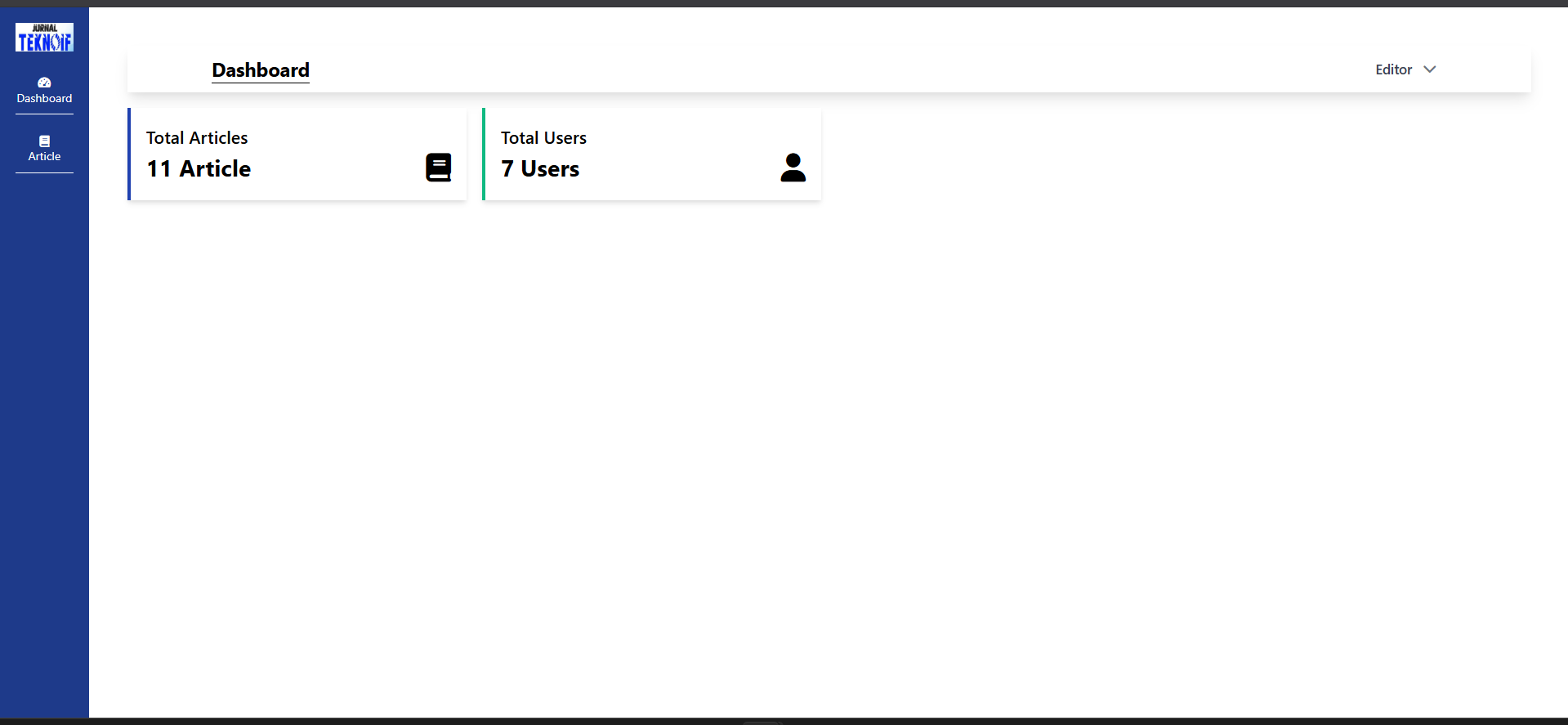
* + - 1. Halaman *Article*



Gambar 23 *Article Chief-Editor*

Gambar 23 merupakan hasil dari perbaikan halaman *articles*. Digunakan untuk menampilkan informasi dari *metadata* *article* yang telah berhasil di proses, dengan perbaikan penambahan informasi *intisari*, jumlah yang di tampikan dalam tabel pada setiap tab dan tombol pencarian.

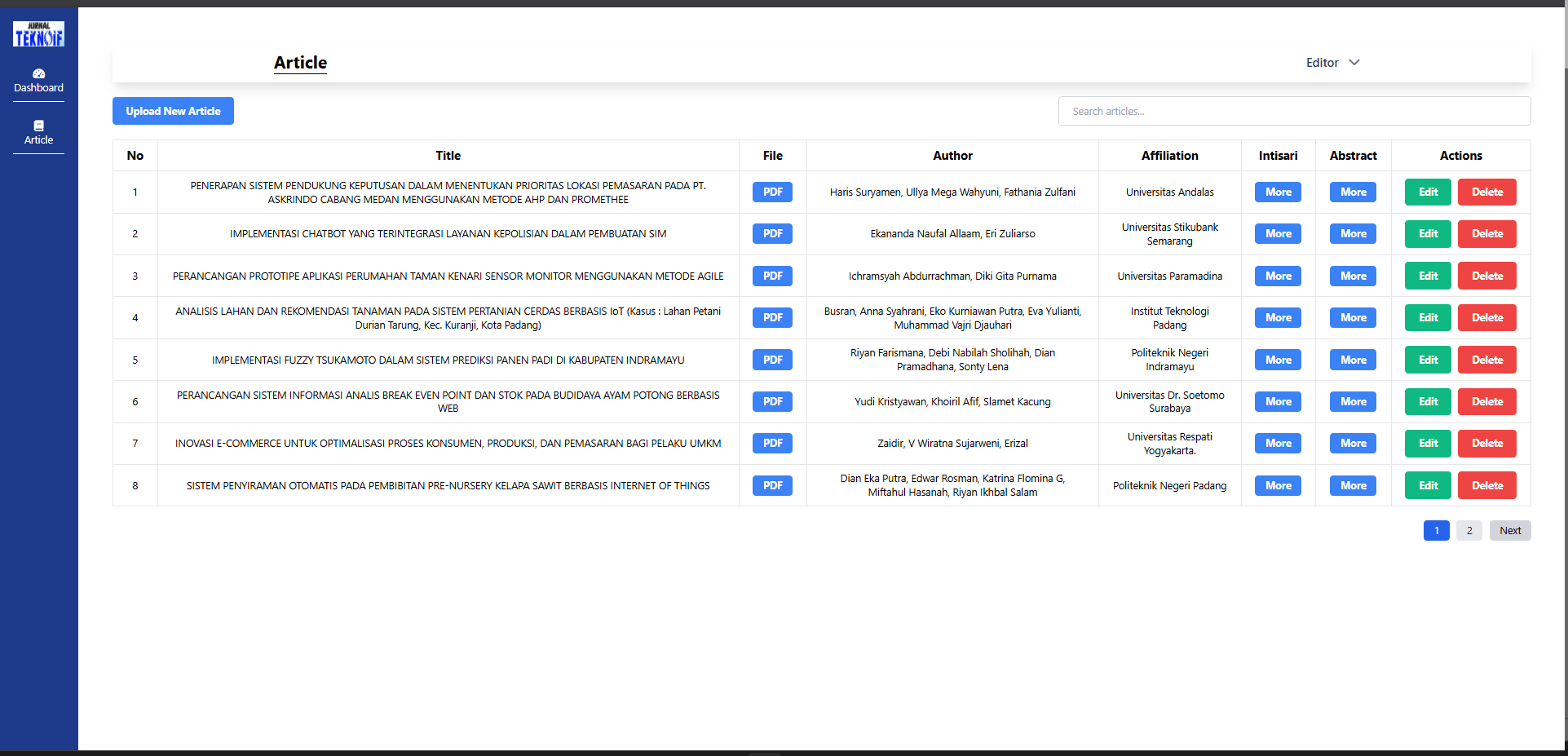
1. Halaman *Editor*
2. Halaman *Dashboard*



Gambar 24 *Dashboard Editor*

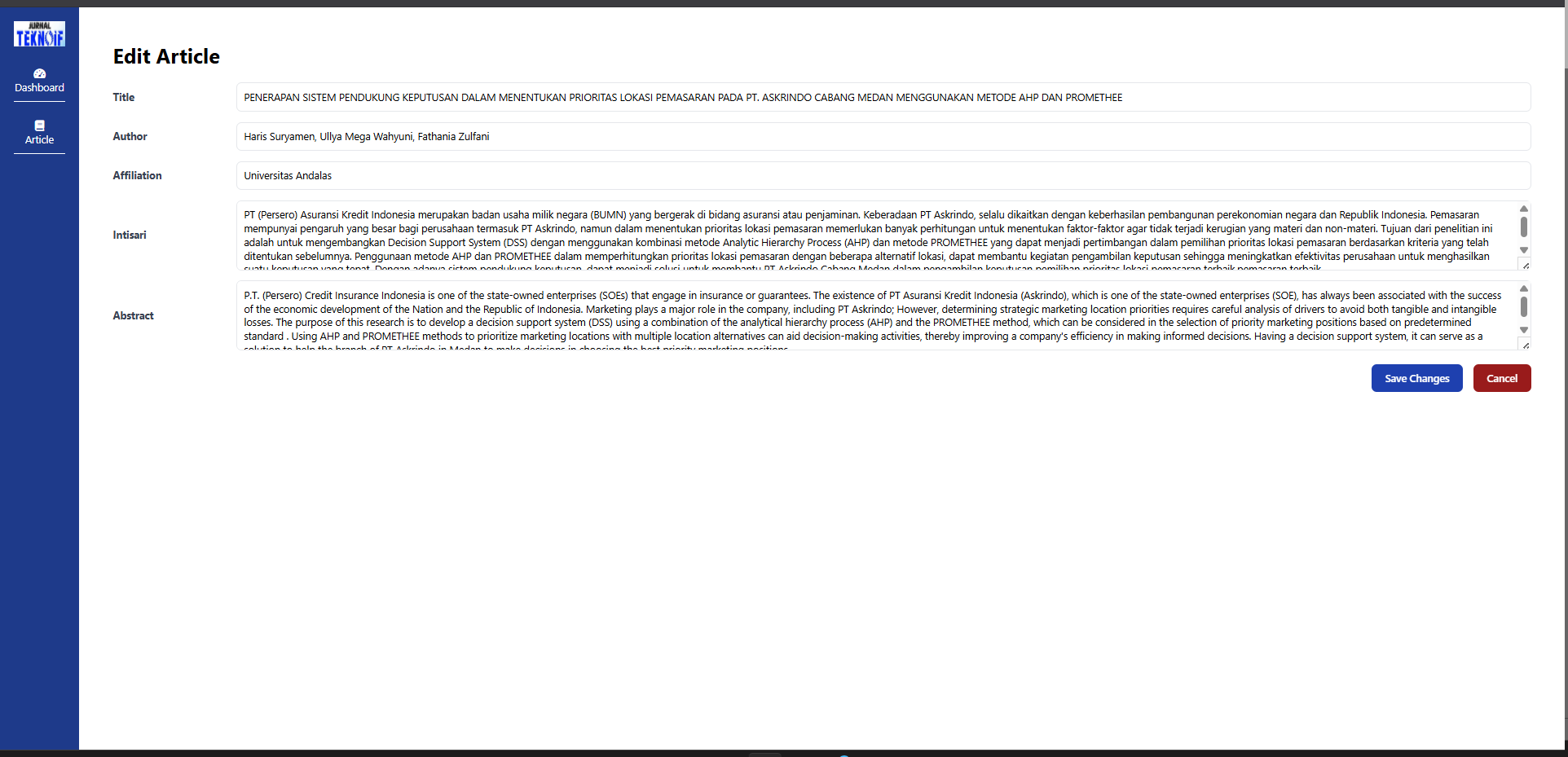
Gambar 24 merupakan hasil perbaikan tampilan dari halaman *dashboad editor* yang dengan perbaikan pada tampilan antar muka dengan penambahan informasi *role* yang sedang aktif.

1. Halaman *Article*



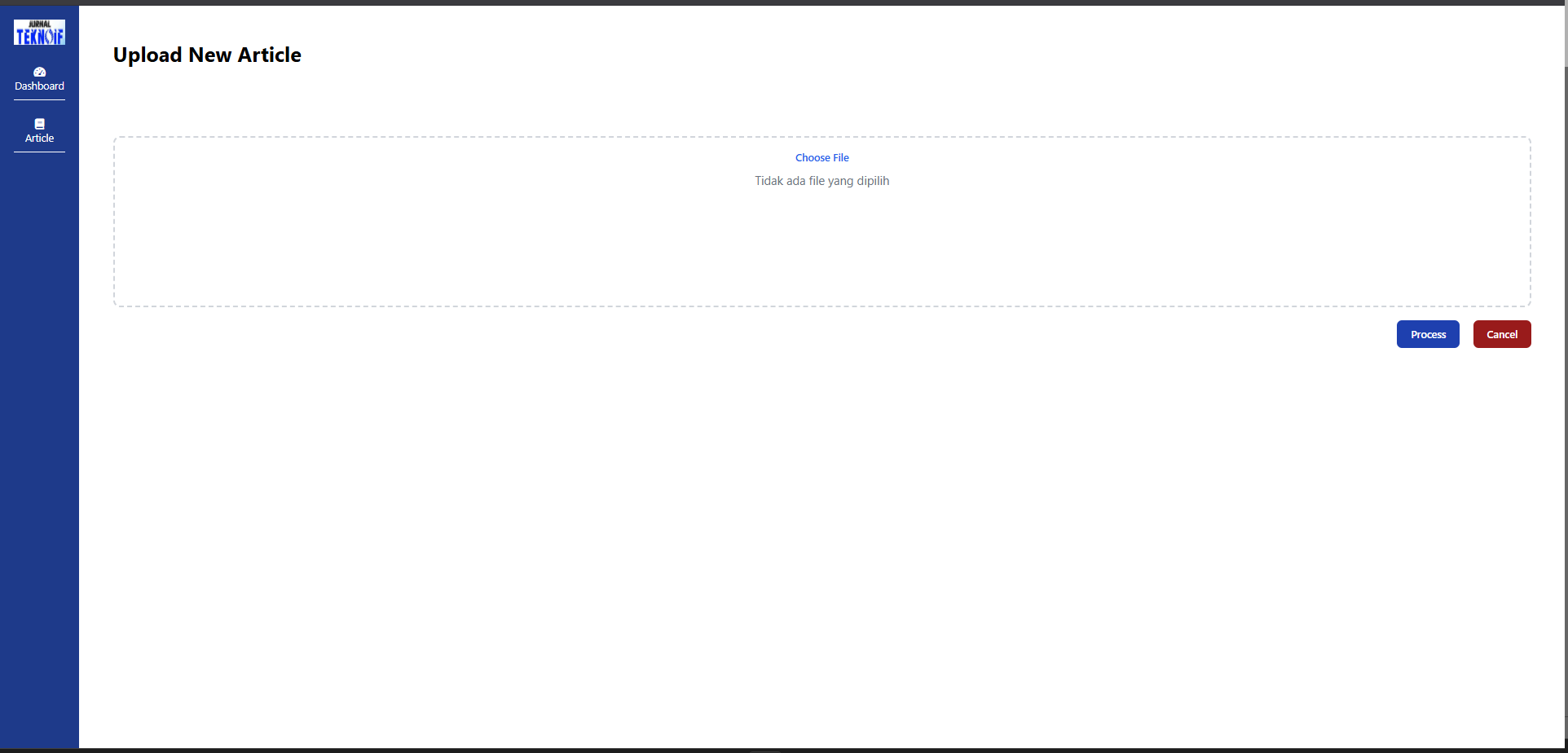
Gambar 25. *Article Editor*

Gambar 25 merupakan hasil perbaikan dengan menampilkan informasi dari *metadata* *article* yang telah berhasil di proses dan perbaikan penambahan informasi *intisari*, jumlah yang di tampikan dalam tabel pada setiap tab tombol pencarian.



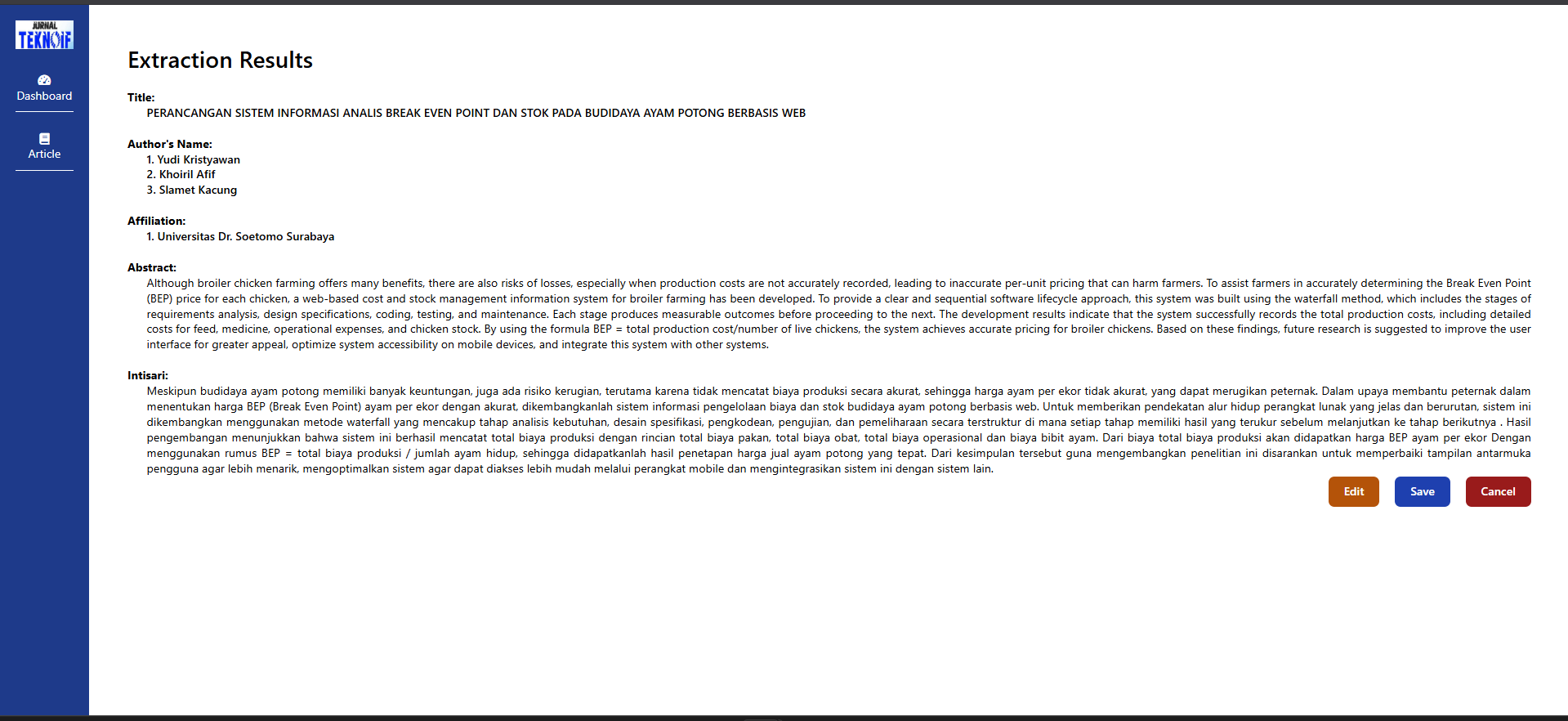
Gambar 26. *Edit Article Editor*

Pada Gambar 26 merupakan tampilan dari *form* edit artikel yang telah selesai di proses dan telah di simpan di sistem. Edit yang dilakukan setalah menyimpan informasi *metadata* artikel di dalam sistem bertujuan untuk memperbaiki kesalahan pada *metadata* tersebut.



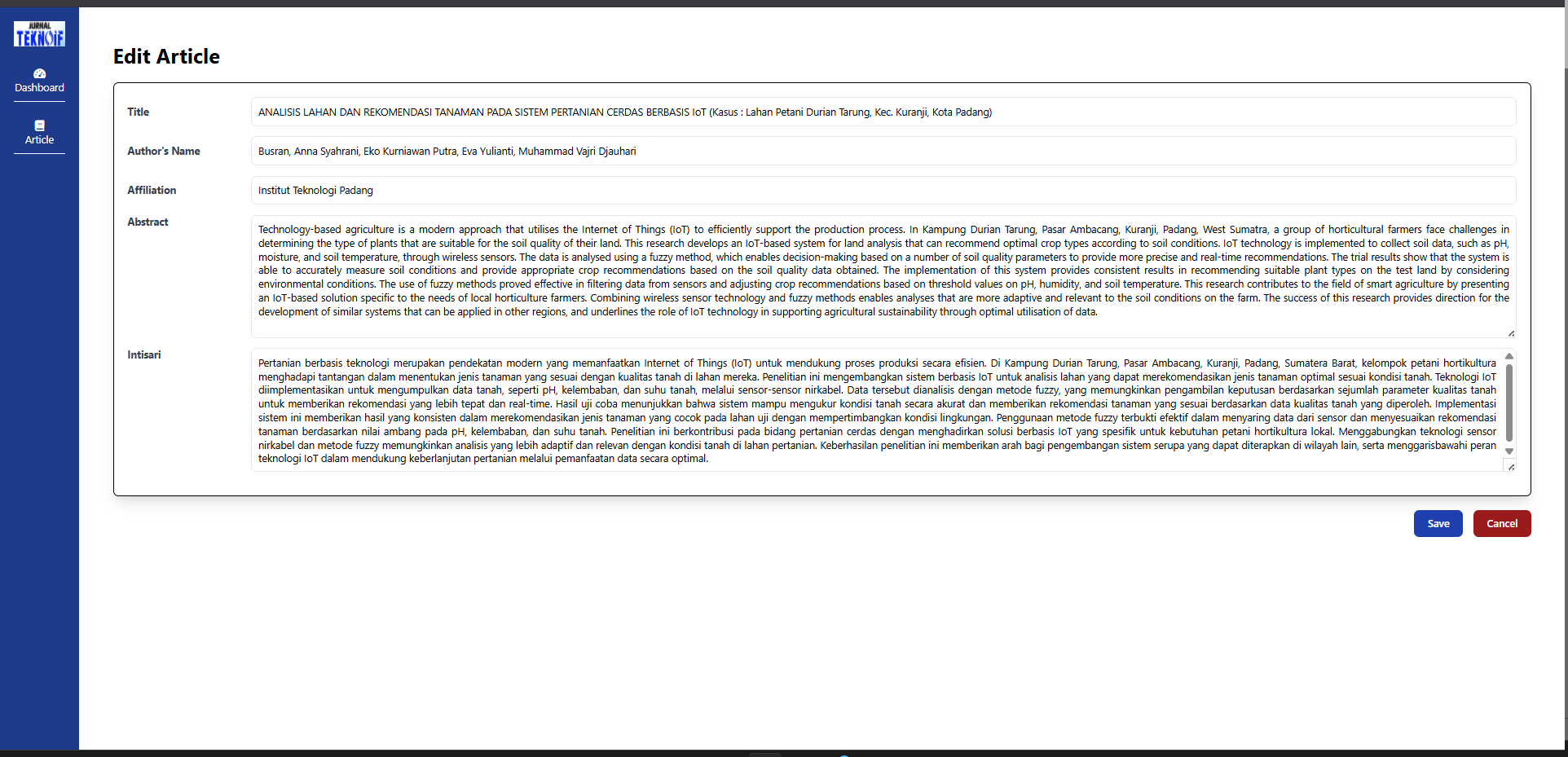
Gambar 27. *Upload Article Editor*

Gambar 27 merupakan tampilan dari *form upload* *article* dan hasil dari perbaikan tampilan *form upload article*. Gambar 27 merupakan langkah awal dengan melakukan *upload article* untuk diproses dan memperoleh informasi *metadata* artikel, sesuai dengan kebutuhan sistem *teknoif*, yaitu judul, nama penulis, afiliasi, dan abstrak.



Gambar 28. *Extraction Results Editor*

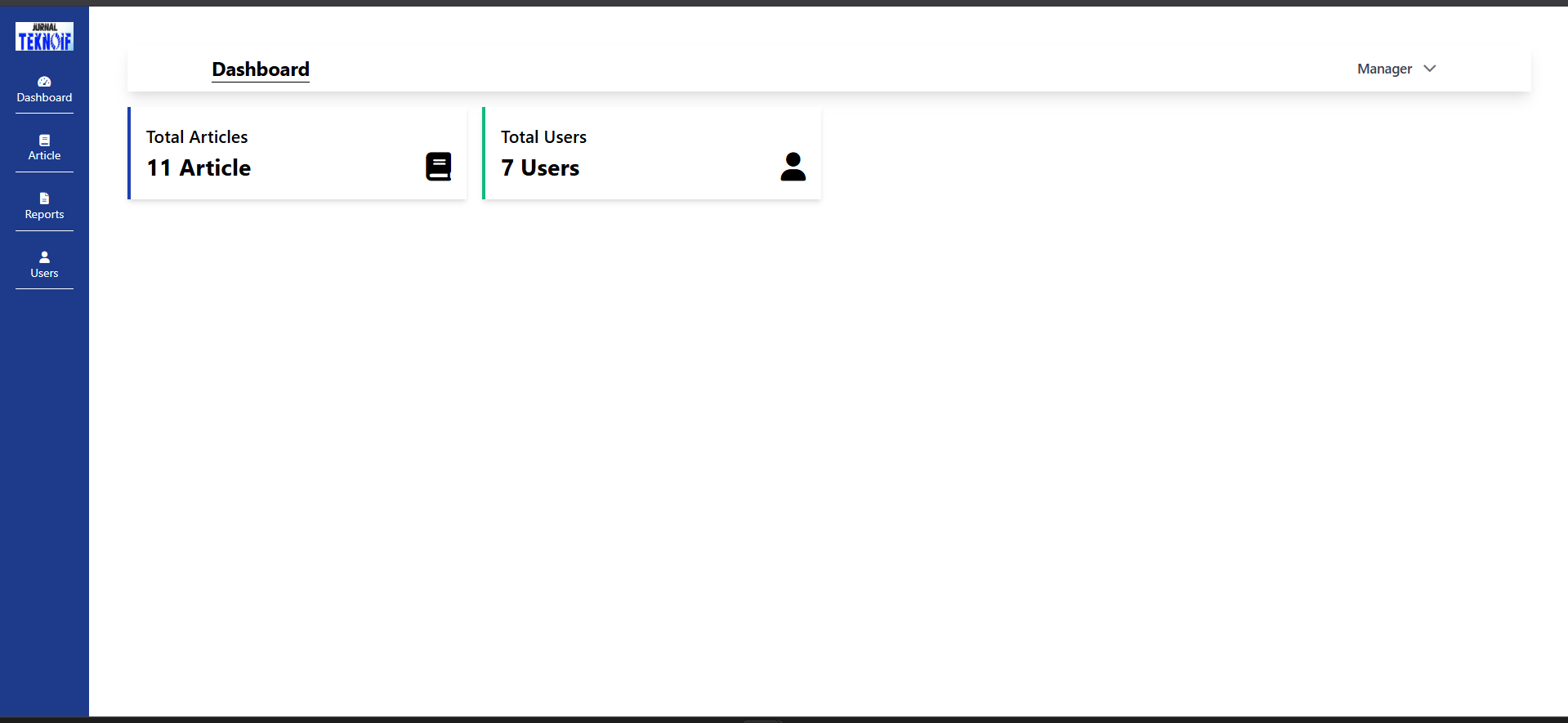
Gambar 28 merupakan hasil perbaikan tampilan dari sebelumnya dengan menampilkan informasi *metadata* artikel, pada hasil pemrosesan tersebut juga terdapat tambahan informasi *intisari* yang dapat di simpan ke dalam *database*



Gambar 29. *Edit Results Extraction Editor*

Gambar 29 merupakan tampilan dari proses edit hasil dari ekstraksi dan pada tampilan baru dari *form* edit yang digunakan untuk menyesuaikan hasil ekstraksi *metadata* artikel jika hasil belum sesuai dengan data yang di inginkan.

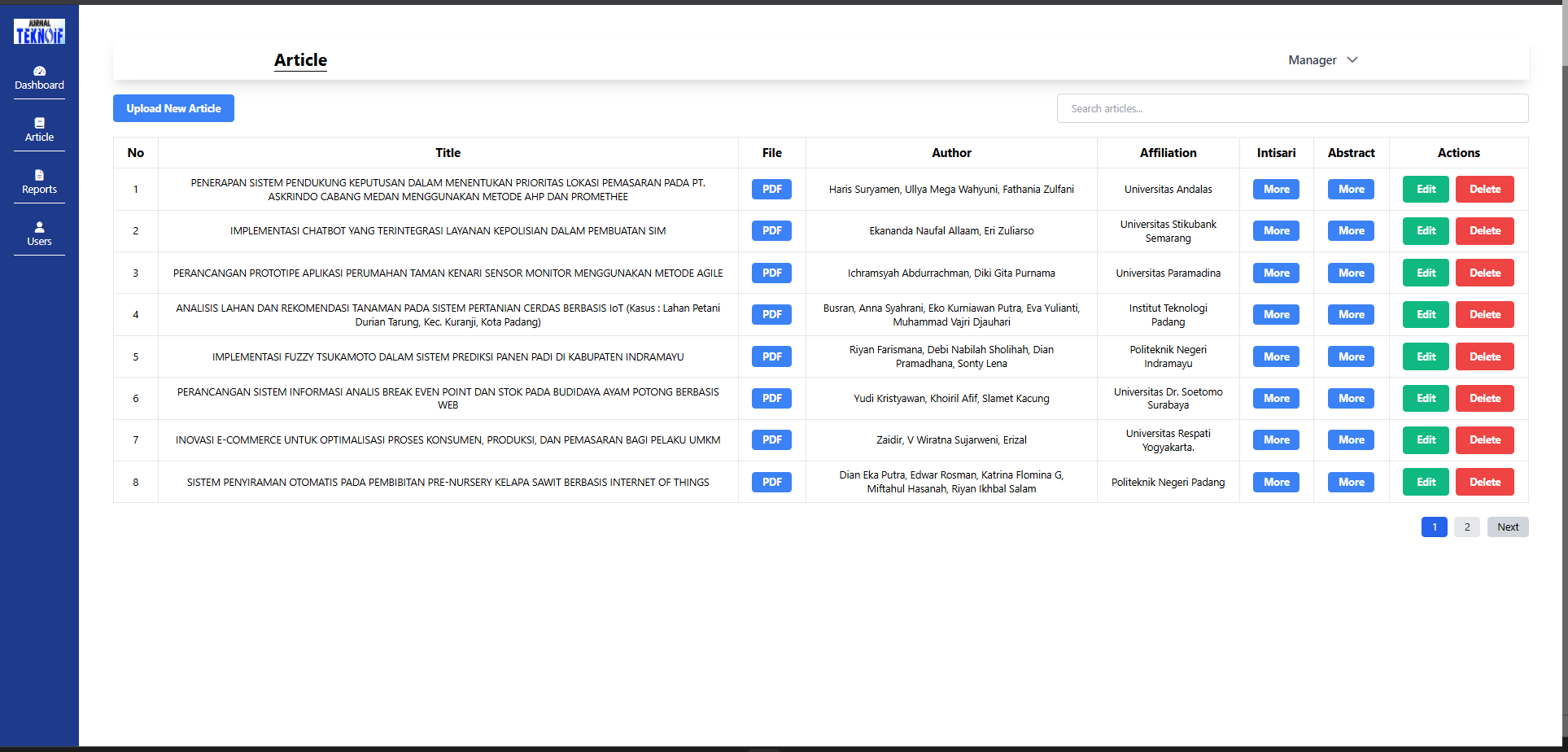
1. Halaman *Manager*
2. Halaman *Dashboard*



Gambar 30. *Dashboad Manager*

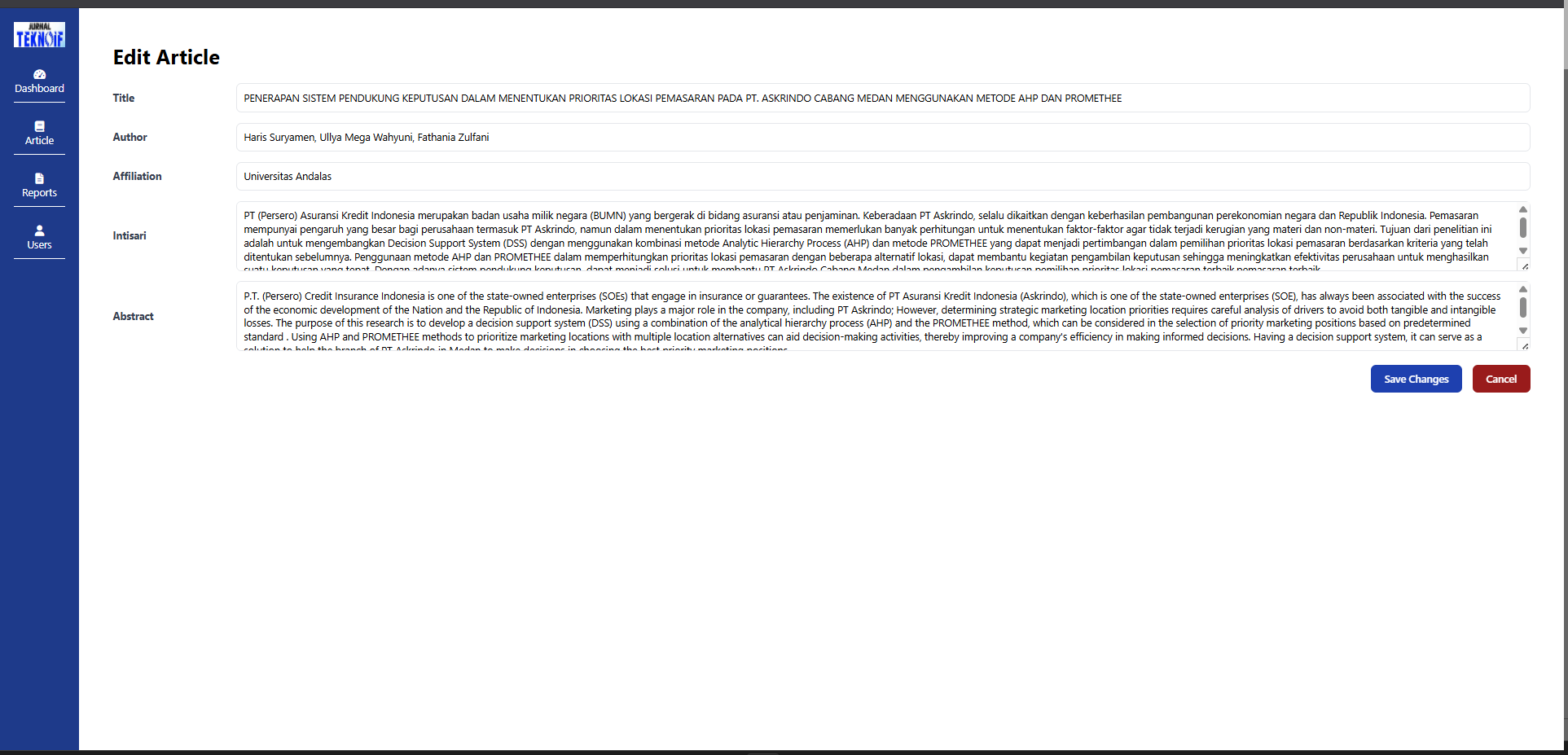
Gambar 30 merupakan tampilan dari halaman *dashboad Manager* yang di perbaiki pada tampilan antar muka dengan penambahan informasi *role* yang sedang aktif.

1. Halaman *Article*



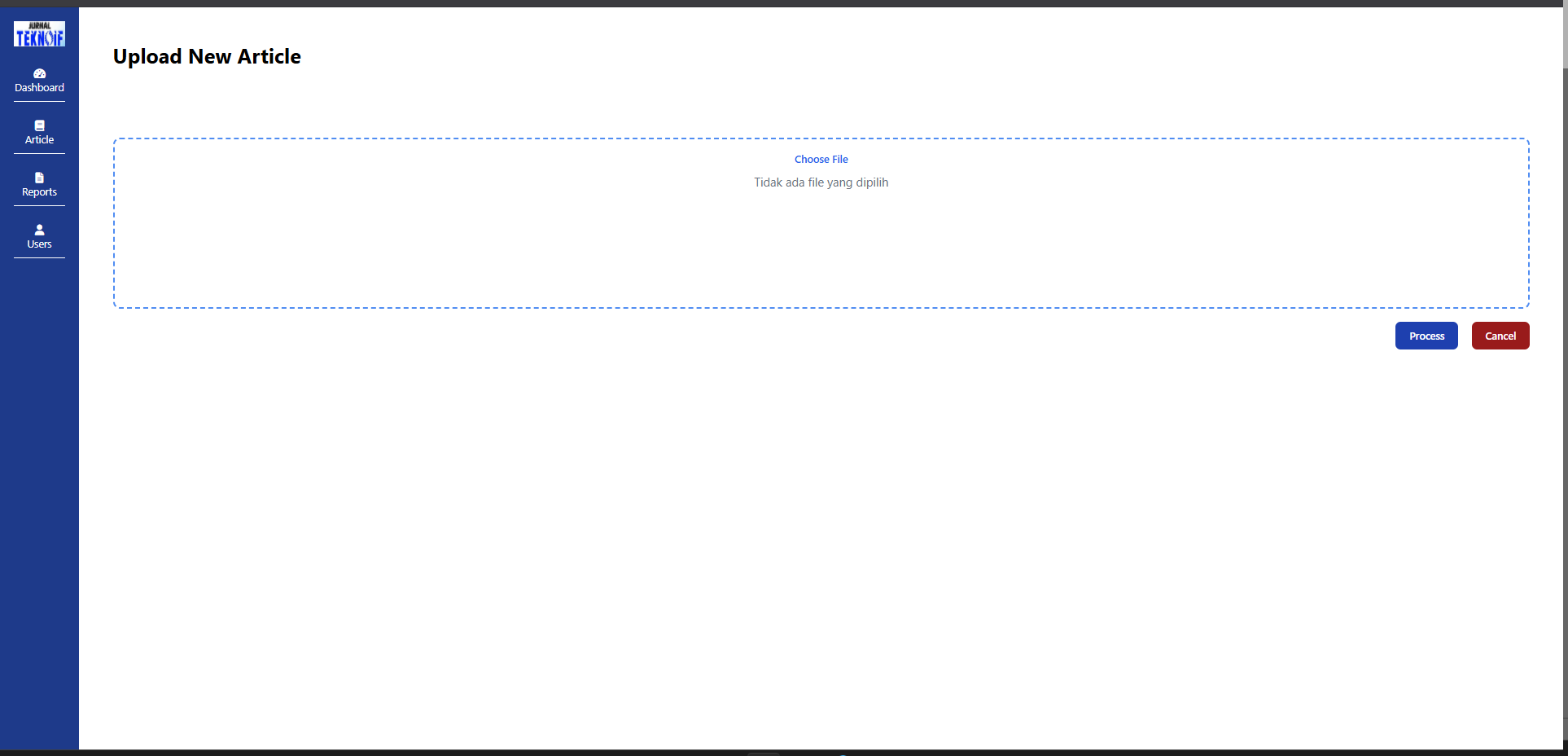
Gambar 31. *Article Manager*

Gambar 31 merupakan tampilan dari halaman *article* dan merupakan hasil perbaikan dengan menampilkan informasi dari *metadata* *article* yang telah berhasil di proses, dengan perbaikan penambahan informasi *intisari*, jumlah yang di tampikan dalam tabel pada setiap tab tombol pencarian.



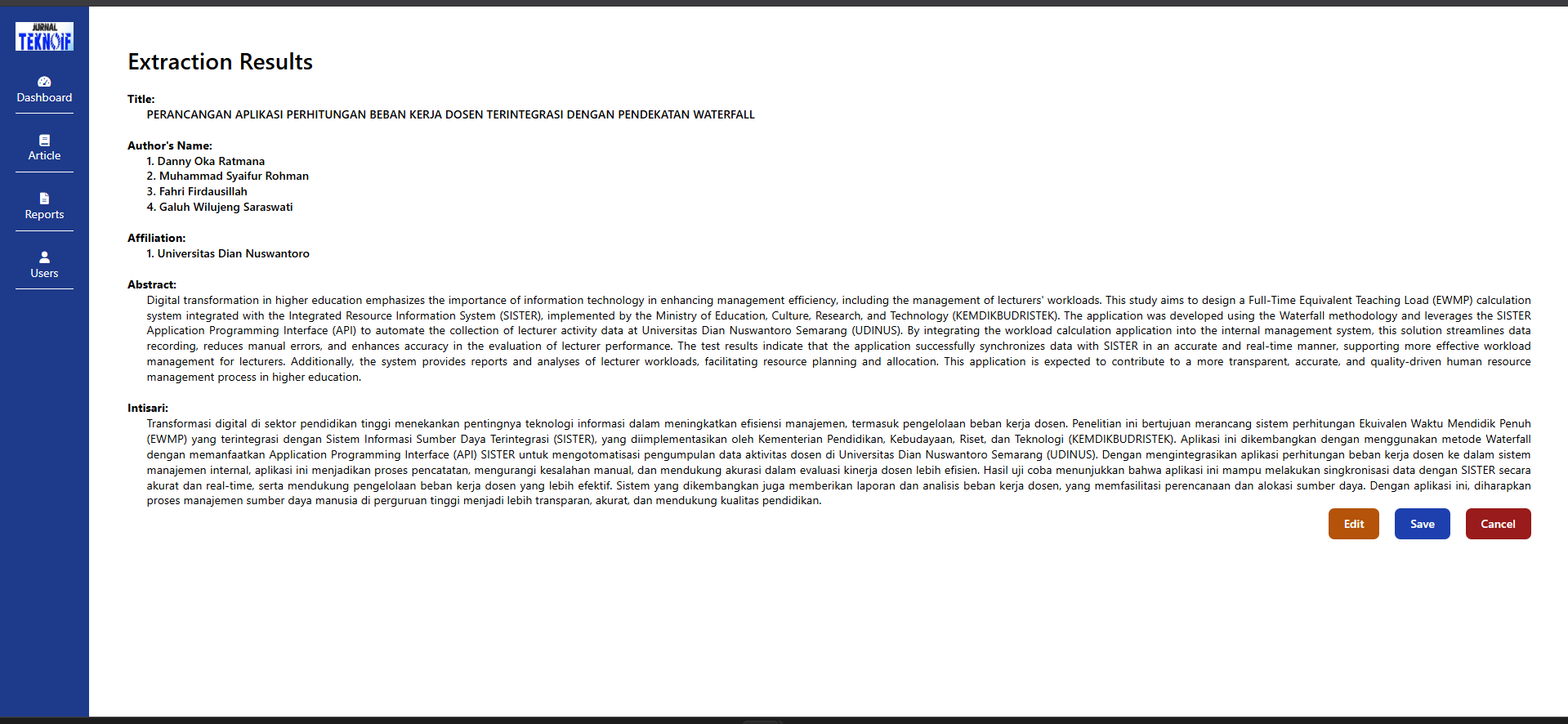
Gambar 32. *Edit Article Manager*

Pada Gambar 32 merupakan tampilan dari *form* edit artikel yang telah selesai di proses dan telah di simpan di sistem. Edit yang dilakukan setalah menyimpan informasi *metadata* artikel di dalam sistem bertujuan untuk memperbaiki kesalahan pada *metadata* tersebut.



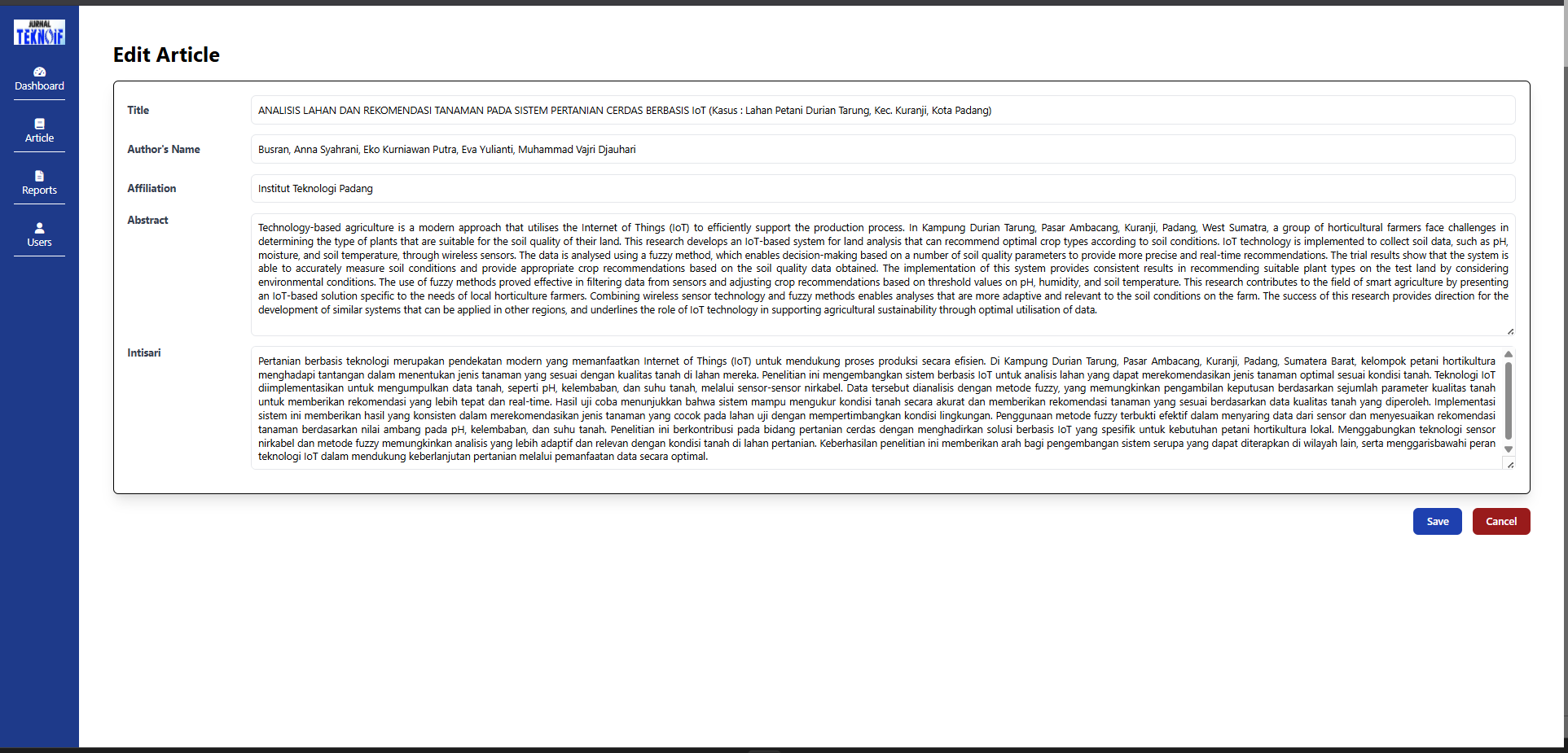
Gambar 33 *Upload New Article Manager*

Pada Gambar 33 ditampilkan tampilan *form upload* artikel yang akan diproses untuk memperoleh informasi *metadata* artikel, sesuai dengan kebutuhan sistem *teknoif*, yaitu judul, nama penulis, afiliasi, dan abstrak.



Gambar 34 *Extraction Results Manager*

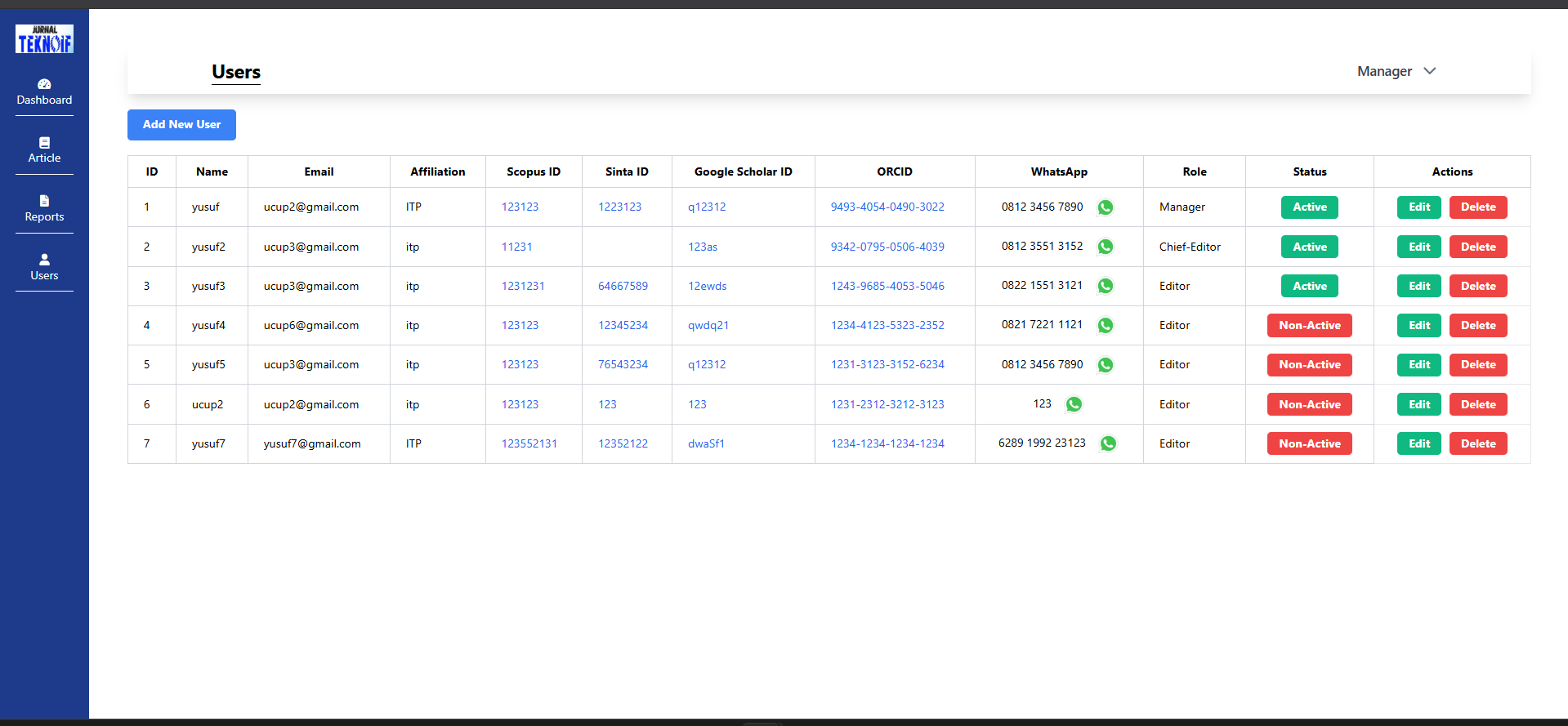
Pada Gambar 34 merupakan hasil *metadata* artikel, pada hasil pemrosesan tersebut terdapat tambahan informasi *intisari* yang dapat di simpan ke dalam database.



Gambar 35. *Edit Process Results Manager*

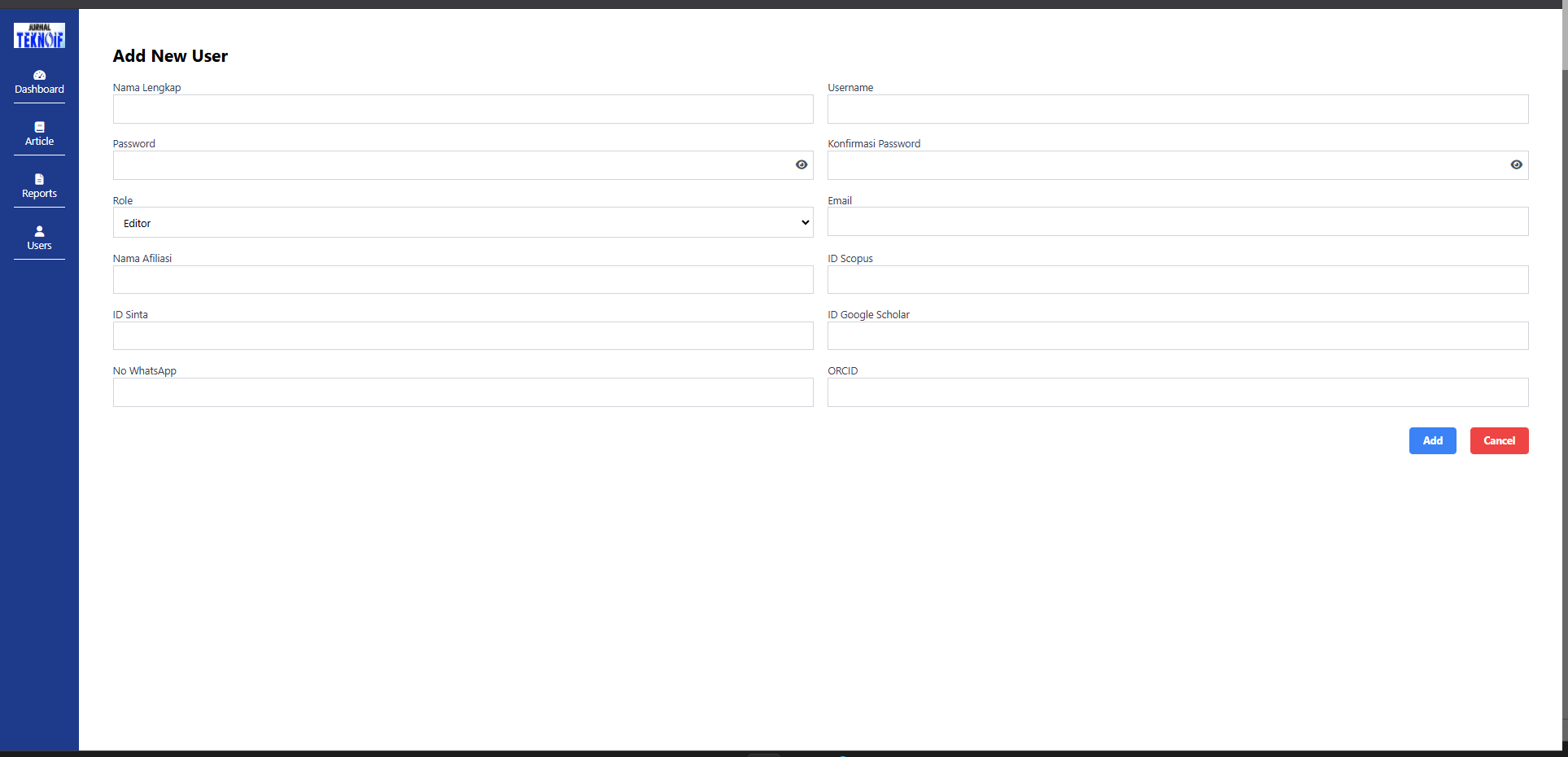
Pada Gambar 35 merupakan *form* edit yang digunakan untuk menyesuaikan hasil ekstraksi *metadata* artikel jika hasil belum sesuai dengan data yang di inginkan.

1. Halaman *Users*



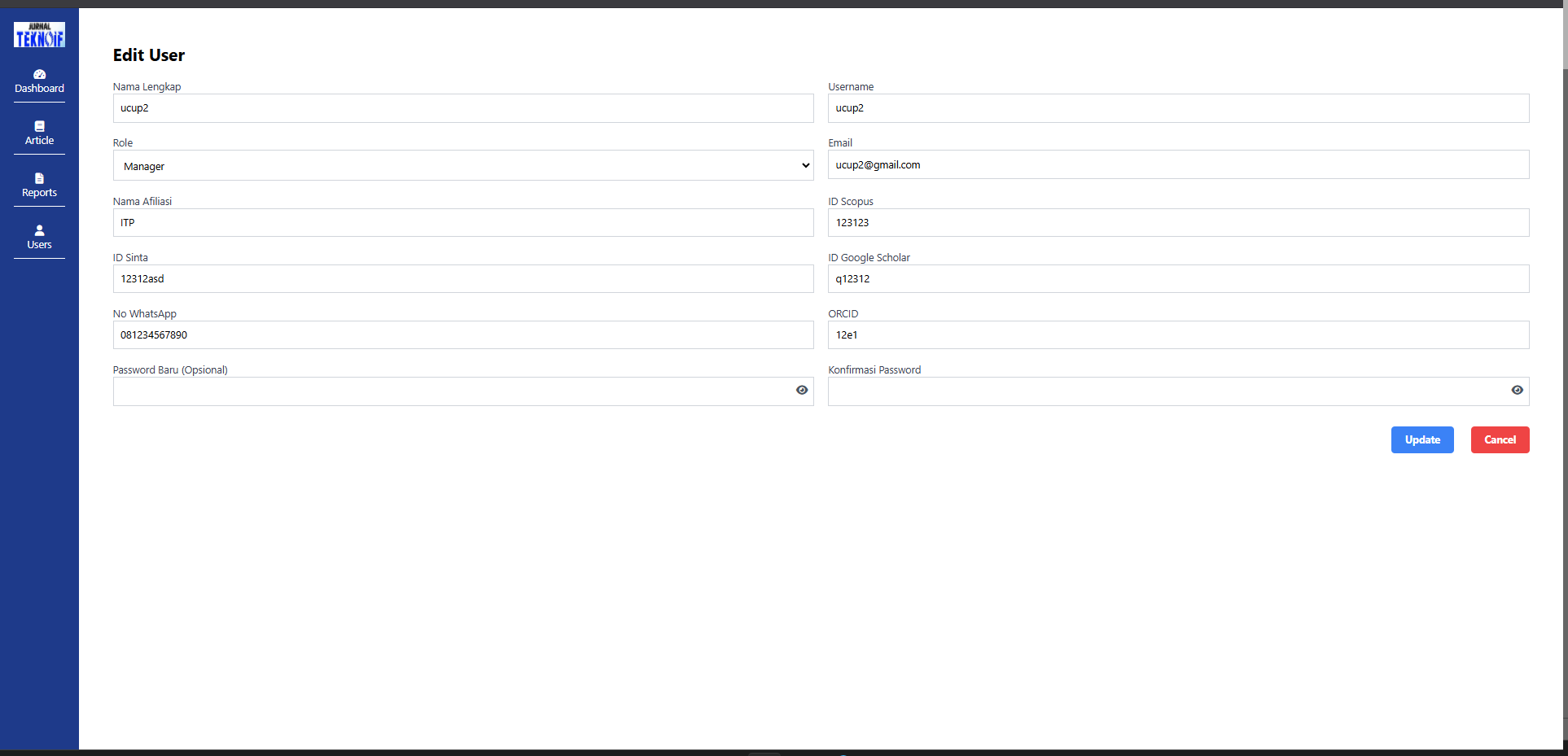
Gambar 36 *Kelola Users Manager*

Pada Gambar 36 merupakan tampilan dari bagian kelola *users* yang telah di perbaiki dengan informasi lebih lengkap dan terdapat informasi serta akses untuk mengaktifkan atau menonaktifkan *user*.



Gambar 37 *Create Users Manager*

Pada Gambar 37 merupakan tampilan dari proses penambahan pengguna baru dalam sistem yang telah di perbaharui dengan data informasi yang di butuhkah lebih lengkap.



Gambar 38 *Update Users Manager*

Pada Gambar 38 merupakan tampilan *update* akun dari pengguna yang telah di perbaharui, proses ini dapat mengubah informasi-informasi dari akun pengguna.

* + 1. Testing

Pada tahap pengujian (*testing*), dilakukan pengujian sistem dengan pengelola jurnal *teknoif* untuk menemukan *bug* atau kesalahan yang mungkin terjadi. Setelah dilakukan pengujian, tidak ditemukan *bug* pada sistem, sehingga sistem dapat dilanjutkan ke tahap berikutnya. Adapun saran dan masukan dari pengelola jurnal *teknoif* dapat dilihat pada Tabel 1.

Tabel 1 Catatan Pengujian Pengguna

| No. | Catatan Pengujian Pengguna |
| --- | --- |
| 1. | *Reports using button in article page* |
| 2. | *Reports* menu *view* grafik *afiliation*, *frist afiliation* |
| 3. | *Add icon pdf* |
| 4. | *icon dashboard* |
| 5. | *Link to sinta, scopus, google scholar and view just code* |

1. PENUTUP
2. Kesimpulan

Berdasarkan hasil perancangan dan implementasi sistem ekstraksi *metadata* artikel menggunakan metode *regular expression (regex)* yang dikembangkan menggunakan metode *Extreme Programming (XP)* dengan dua iterasi, dapat disimpulkan bahwa sistem mampu menjalankan fungsinya dengan baik. Proses ekstraksi dilakukan melalui tiga tahap utama, yaitu ekstraksi teks dari *file* *pdf*, pembersihan data *(cleaning)* dengan tiga kriteria, dan ekstraksi *metadata* menggunakan pola *regex* yang mencakup empat kriteria. Sistem ini berhasil mengekstraksi informasi penting seperti judul, nama penulis, afiliasi, dan abstrak dari dokumen *pdf* secara otomatis. Seluruh proses berjalan sesuai dengan fungsionalitas yang diharapkan dan telah berhasil diuji menggunakan metode *black-box* testing tanpa ditemukan *bug* yang dapat mengganggu kinerja sistem. Dengan demikian, sistem ini dapat membantu dalam pengelolaan arsip jurnal *teknoif*.

1. Saran

Berdasarkan pembahasan yang telah dilakukan, terdapat beberapa saran untuk pengembangan sistem ke depannya, yaitu:

1. Mengoptimalkan sistem dalam proses ekstraksi *metadata* artikel agar dapat berjalan secara optimal tidak hanya pada *layout* atau tata letak artikel Jurnal *teknoif*, tetapi juga pada berbagai format jurnal lainnya yang memiliki variasi struktur yang berbeda.
2. Menambahkan fitur yang dapat memberikan informasi atau menetapkan pihak yang bertanggung jawab dalam memproses artikel, guna mendukung manajemen alur kerja dan distribusi tugas secara lebih terstruktur.
3. DAFTAR PUSTAKA

[1] F. Reza, I. K. D. Indah, and M. Ropianto, “Perancangan Dan Implementasi Institutional Repository Dengan Metadata Dublin Core,” *komtekinfo*, pp. 125–132, Dec. 2022, doi: 10.35134/komtekinfo.v9i4.318.

[2] F. Rahutomo, D. A. Irawati, and M. A. E. Pramudita, “Pengembangan Sistem Ekstraksi Metadata Artikel ilmiah secara Otomatis,” *JTIIK*, vol. 6, no. 2, pp. 123–128, Feb. 2019, doi: 10.25126/jtiik.2019621227.

[3] Y. Puspitarani, Y. Syukriyah, and U. Widyatama, “Pemanfaatan Optical Character Recognition Dan Text Feature Extraction Untuk Membangun Basisdata Pengaduan Tenaga Kerja,” vol. 4, no. 4, 2020.

[4] D. N. Fadhillah and A. Rachman, “IMPLEMENTASI REGEX PADA PEMBERIAN KOMENTAR KODE PROGRAM HTML,” *JARS*, vol. 2, no. 1, pp. 1–5, Dec. 2023, doi: 10.24929/jars.v2i1.3078.

[5] J. M. Bintang, M. F. Ashshidiq, and H. F. Dzakwan, “Penerapan Algoritma String Matching dan Regular Expression pada Aplikasi Kamus Besar Bahasa Indonesia (KBBI),” *bios*, vol. 4, no. 1, pp. 34–41, Mar. 2023, doi: 10.37148/bios.v4i1.57.

[6] N. A. Septiani and L. D. Yanti, “SISTEM INFORMASI PEMASANGAN IKLAN KORAN PADA PT. HARIAN TOPSKOR DENGAN METODE EXTREME PROGRAMMING (XP),” vol. 06, Dec. 2021.

[7] F. A. T. Tobing and R. Nainggolan, “ANALISIS PERBANDINGAN PENGGUNAAN METODE BINARY SEARCH DENGAN REGULAR SEARCH EXPRESSION,” *jmika*, vol. 4, no. 2, pp. 168–172, Oct. 2021, doi: 10.46880/jmika.Vol4No2.pp168-172.

[8] I. Ahmad, R. I. Borman, J. Fakhrurozi, and G. G. Caksana, “Software Development Dengan Extreme Programming (XP) Pada Aplikasi Deteksi Kemiripan Judul Skripsi Berbasis Android,” vol. 5, no. 2, 2020.

[9] Q. E. Fazrin, T. Lisnawati, S. Nurhayati, J. B. Satya, and D. Alamsyah, “Penerapan Metode Pengembangan Sistem Extreme Programing (XP) Pada Aplikasi Presensi Karyawan dengan QR Code,” *bits*, vol. 3, no. 3, pp. 164–170, Dec. 2021, doi: 10.47065/bits.v3i3.1018.

[10] R. I. Borman, A. T. Priandika, and A. R. Edison, “Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan,” vol. 8, no. 3, 2020.

[11] K. Nistrina and L. Sahidah, “UNIFIED MODELLING LANGUAGE (UML) UNTUK PERANCANGAN SISTEM INFORMASI PENERIMAAN SISWA BARU DI SMK MARGA INSAN KAMIL,” *Jurnal Sistem Informasi*, vol. 04, 2022.

[12] B. J. M. Putra, A. Fu’adi, and D. A. F. Yuniarti, “Analisa dan Rancangan Sistem Informasi Pariwisata Pacitan dengan UML dan ERD,” vol. 7, no. 1, 2022.

[13] K. Afiifah, Z. Fira Azzahra, and A. Dwi Anggoro, “Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review,” *INTECH (Informatika dan Teknologi)*, vol. 3, no. 1, pp. 8–11, Apr. 2022, doi: 10.54895/intech.v3i1.1261.

[14] F. Azmi, “SISTEM PARKIR DENGAN RADIO FREQUENCY IDENTIFICATION BERBASIS GELOMBANG RADIO,” vol. 3, 2023.

[15] Ni Nyoman Emang Smrti, A. I Putu Gd Sukenada, D. T. R. Ni Kadek, A. Adnan, and J. Pande Putu Ode, “Flowgorithm Sebagai Penunjang Pembelajaran Algoritma dan Pemrograman,” *bangkitindonesia*, vol. 12, no. 1, pp. 56–64, Mar. 2023, doi: 10.52771/bangkitindonesia.v12i1.218.

[16] Z. Tuasamu *et al.*, “Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan DFD dan Flowchart Pada Bisnis Porobico,” *Jurnal Bisnis dan Manajemen (JURBISMAN)*, vol. 1, no. 2, Art. no. 2, May 2023, doi: 10.61930/jurbisman.v1i2.181.