

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisis Sistem

*Framework web* seperti React JS, Laravel, dan Bootstrap memainkan peran penting dalam pengembangan aplikasi berbasis *web*. Ketiganya memiliki fungsi dan keunggulan yang berbeda-beda, mulai dari kemudahan membangun antarmuka pengguna interaktif, efisiensi dalam pengelolaan *back-end*, hingga penyediaan komponen siap pakai untuk *front-end*. Pemilihan ketiga *framework* ini didasarkan pada keunggulan masing-masing dalam mendukung pengembangan sistem yang efisien, terstruktur, dan responsif.

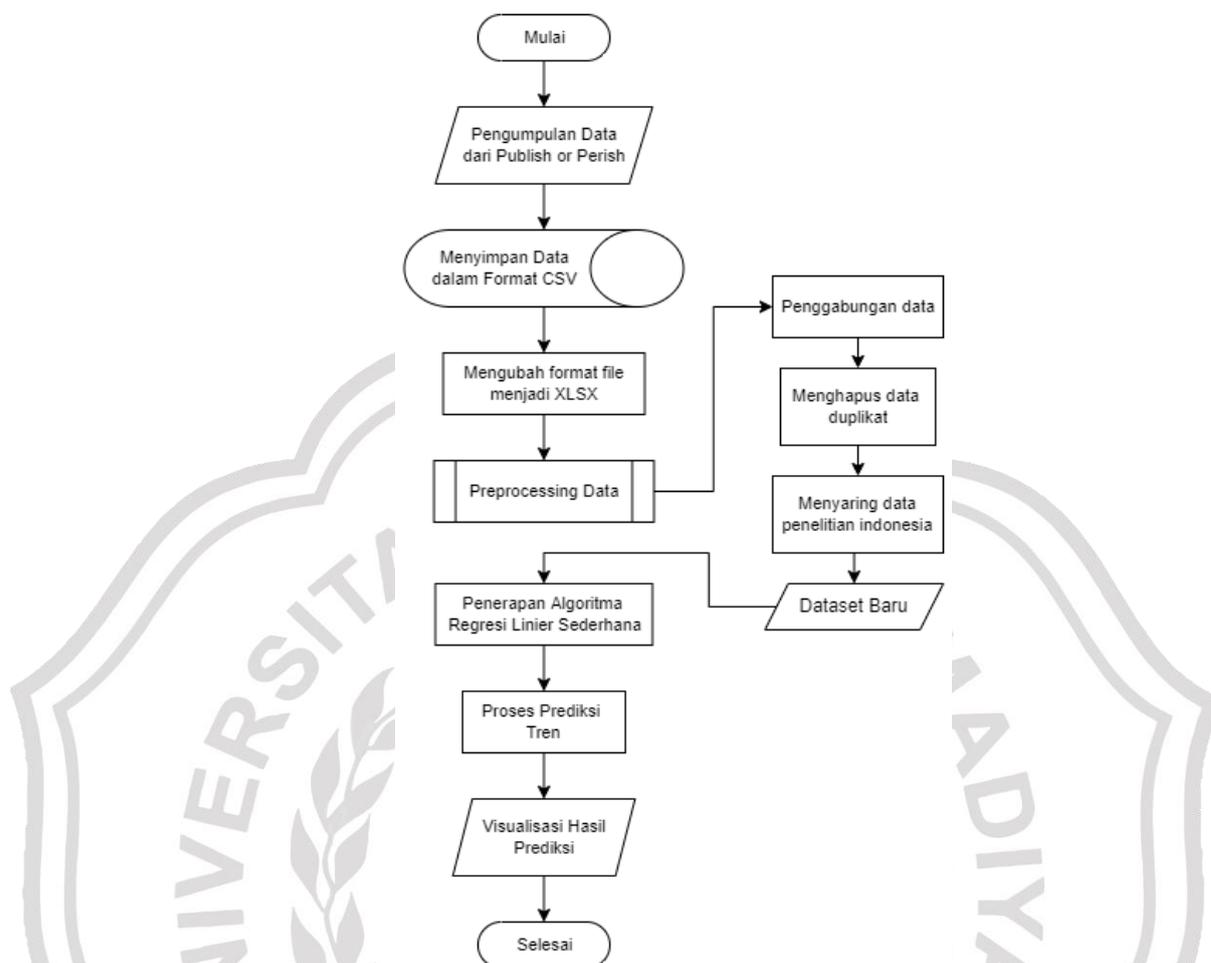
Memahami arah tren penggunaan *framework* menjadi hal yang penting di tengah pesatnya perkembangan teknologi *web* ini. Sayangnya, masih sedikit kajian yang meneliti seberapa besar *framework* tertentu digunakan dalam dunia akademis. Padahal, informasi tersebut dapat membantu para pengembang, pendidik, dan peneliti untuk menentukan teknologi yang sesuai dengan kebutuhan. Ketika arah perkembangan tidak terlihat dengan jelas, risiko pemilihan *framework* yang kurang tepat dalam penelitian maupun pengembangan sistem akan meningkat.

Melalui analisis tren *framework*, dapat diperoleh wawasan mengenai sejauh mana *framework* digunakan dan diminati dalam penelitian. Selain itu, melalui proses prediksi tren, dapat memberikan gambaran mengenai potensi perkembangan penggunaan *framework* di masa mendatang berdasarkan pola data historis yang tersedia.

#### 3.2 Hasil Analisis Sistem

Proses analisis tren dalam penelitian ini dilakukan berdasarkan data publikasi artikel ilmiah yang dikumpulkan melalui aplikasi *Publish or Perish* pada *web* Google Scholar. Data yang dikumpulkan mencakup artikel-artikel yang memuat kata kunci atau judul yang mengandung nama

*framework* yang menjadi objek penelitian, yaitu React JS, Laravel, dan Bootstrap. Data artikel yang diperoleh akan digunakan untuk melihat perkembangan penggunaan *framework* React JS, Laravel, dan Bootstrap dalam kurun waktu 20 tahun terakhir yakni tahun 2005 - 2024. Proses pengumpulan data menggunakan aplikasi *Publish or Perish* dari *web* Google Scholar menghasilkan sejumlah data mentah untuk masing-masing *framework* yang menjadi objek penelitian. Data dikumpulkan dalam dua kategori, yaitu berdasarkan pencarian pada judul artikel (*title*) dan berdasarkan kata kunci (*keyword*). Secara keseluruhan, proses pengumpulan data menghasilkan sebanyak 68.956 artikel yang berkaitan dengan *framework* React JS, Laravel, dan Bootstrap. Data-data tersebut akan melalui tahap *preprocessing* sebelum digunakan dalam proses analisis dan prediksi tren, hal ini dilakukan untuk memastikan hanya data artikel yang relevan yang digunakan. Proses ini mencakup penggabungan data dari hasil pencarian berdasarkan judul dan kata kunci, penghapusan data duplikat, serta identifikasi artikel yang sesuai dengan konteks penelitian di Indonesia. Setelah melalui proses *preprocessing*, data kemudian dibagi menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*). Data latih mencakup artikel dari tahun 2005 hingga 2020 dan digunakan untuk membentuk model regresi, sedangkan data uji mencakup artikel dari tahun 2021 hingga 2024 yang digunakan untuk melakukan prediksi menggunakan model yang telah dibentuk. Pembagian ini dilakukan untuk mengevaluasi seberapa baik model dapat memprediksi tren berdasarkan data historis. Untuk memberikan gambaran umum mengenai tahapan proses dalam penelitian ini, berikut ditampilkan alur sistem yang diterapkan dalam penelitian ini pada Gambar 3.1.



**Gambar 3.1** Flowchart Sistem Prediksi Tren Penggunaan *Framework*

Berikut adalah penjelasan dari masing-masing proses yang terdapat pada *Flowchart* Sistem Prediksi Tren Penggunaan *Framework* pada Gambar 3.1 :

1. Pengumpulan Data dari *Publish or Perish*

Data dikumpulkan menggunakan aplikasi *Publish or Perish* dengan sumber pencarian dari Google Scholar. Data yang dicari adalah publikasi artikel ilmiah yang berkaitan dengan *framework* React JS, Laravel, dan Bootstrap dalam kurun waktu tahun 2005 hingga 2024.

2. Menyimpan Data dalam Format CSV

Hasil pengumpulan data dari *Publish or Perish* kemudian disimpan dalam format file CSV.

### 3. Mengubah Format File menjadi XLSX

File dengan format CSV diubah menjadi format Excel (.xlsx) agar lebih kompatibel dengan proses pengolahan data yang dilakukan menggunakan Microsoft Excel maupun perangkat lunak lainnya.

### 4. *Preprocessing* Data

Pada tahap ini dilakukan penyaringan data yang terdiri dari beberapa langkah :

- a. Penggabungan Data : Data dari pencarian *title* dan *keyword* digabungkan menjadi satu.
- b. Menghapus Data Duplikat : Setelah penggabungan, dilakukan penghapusan data yang identik agar tidak terjadi pengulangan data.
- c. Menyaring Data Penelitian Indonesia : Data yang tidak relevan dengan penelitian Indonesia akan dihapus.
- d. Dataset Baru : Hasil setelah dilakukan *preprocessing* adalah dataset baru yang siap digunakan untuk proses selanjutnya.

### 5. Penerapan Algoritma Regresi Linier Sederhana

Dataset baru kemudian dibagi menjadi dua bagian, yaitu data latih (tahun 2005–2020) dan data uji (tahun 2021–2024). Pada tahap ini, data latih digunakan untuk melakukan perhitungan dan membentuk model persamaan regresi linier sederhana untuk masing-masing *framework*.

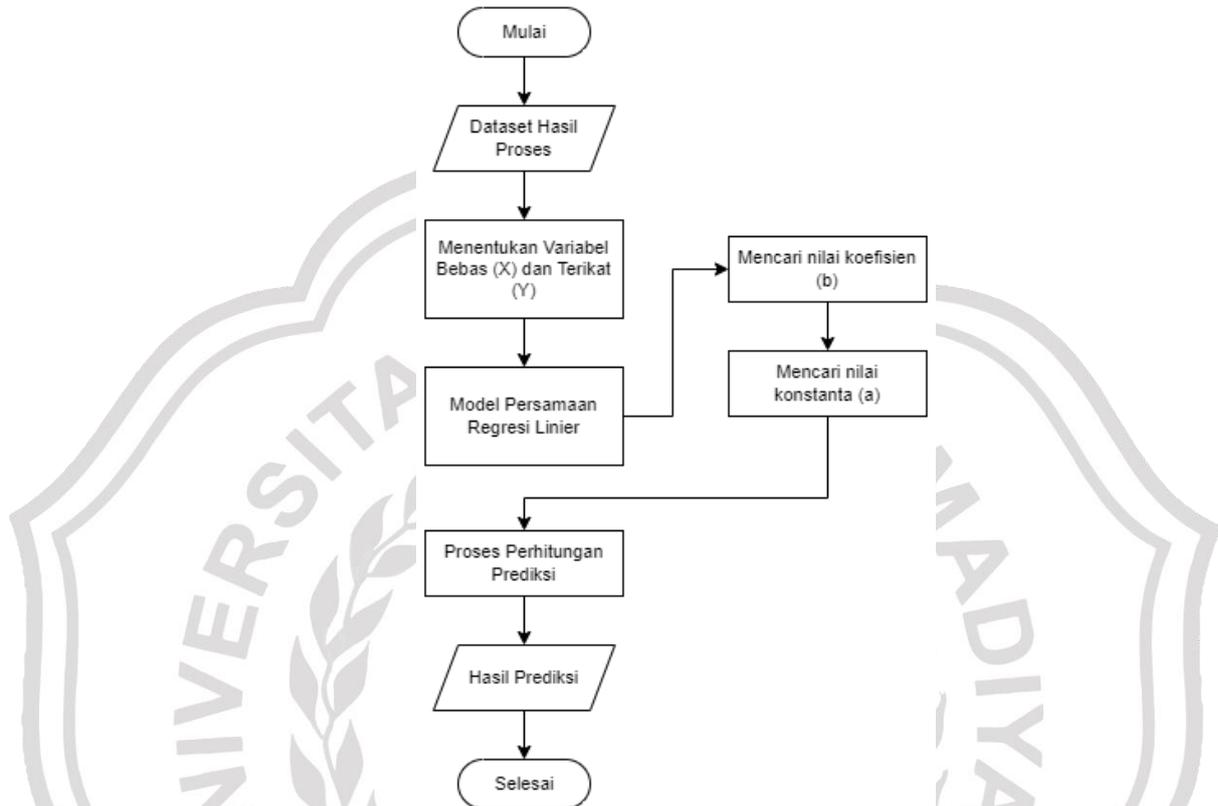
### 6. Proses Prediksi Tren

Model persamaan regresi yang telah dibentuk kemudian digunakan untuk memprediksi jumlah publikasi *framework* pada tahun 2021 hingga 2024.

### 7. Visualisasi Hasil Prediksi

Hasil prediksi kemudian disajikan dalam bentuk tabel atau grafik agar lebih mudah dipahami oleh pengguna.

Selanjutnya, untuk menggambarkan proses perhitungan prediksi tren lebih spesifik, digunakan *flowchart* penerapan metode Regresi Linier Sederhana yang ditampilkan pada Gambar 3.2.



**Gambar 3. 2** *Flowchart* Proses Regresi Linier Sederhana

Gambar 3.2 menampilkan alur proses penerapan metode Regresi Linier Sederhana dalam penelitian ini. Berikut adalah penjelasan setiap prosesnya :

#### 1. Dataset Hasil Proses

Memasukkan dataset hasil pengolahan data dari tahap *preprocessing*. Dataset ini berisi dua kolom utama: tahun dan jumlah publikasi. Dataset kemudian dibagi menjadi dua bagian: data latih (tahun 2005–2020) digunakan untuk membentuk model, data uji (tahun 2021–2024) digunakan untuk melakukan prediksi dan perbandingan.

## 2. Menentukan Variabel Bebas dan Terikat

Proses ini bertujuan untuk mengidentifikasi variabel yang digunakan. Dalam penelitian ini, variabel bebas (X) adalah tahun, dan variabel terikat (Y) adalah jumlah publikasi *framework* pada tahun tersebut.

## 3. Model Persamaan Regresi Linier

Dengan menggunakan data latih, dibentuk model persamaan regresi linier sederhana untuk masing-masing *framework*, yang merepresentasikan hubungan antara variabel X dan Y.

### a. Mencari Nilai Koefisien (b)

Koefisien (b) menunjukkan seberapa besar perubahan pada variabel Y yang dihasilkan oleh setiap unit perubahan pada variabel X.

### b. Mencari Nilai Konstanta (a)

Setelah mendapatkan nilai koefisien, selanjutnya dihitung nilai konstanta (a) atau *intercept*. Nilai ini merepresentasikan nilai Y ketika X bernilai nol.

## 4. Proses Perhitungan Prediksi

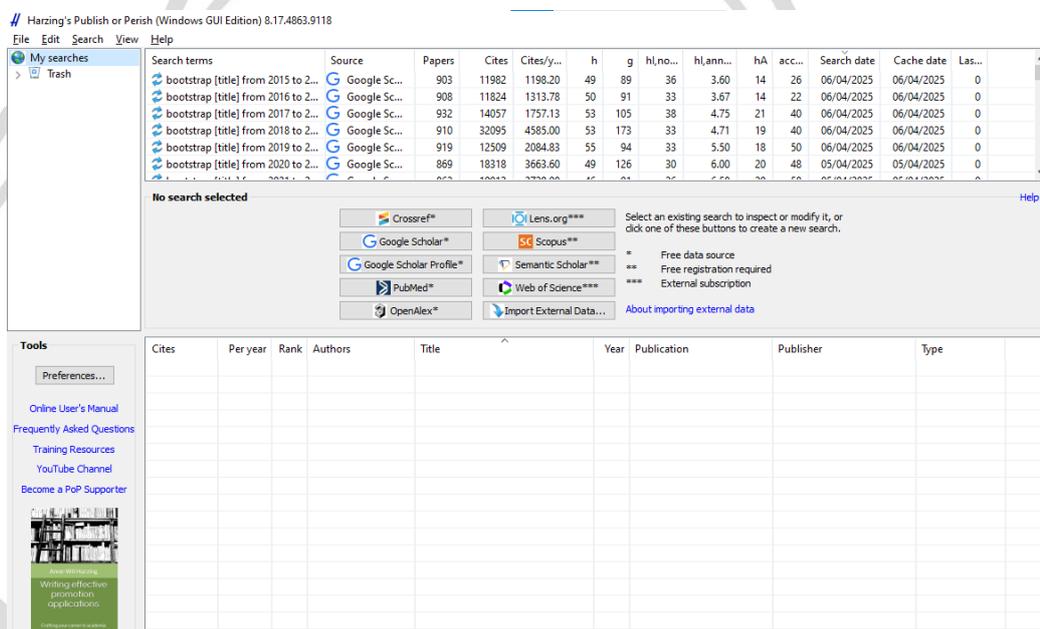
Dengan menggunakan model persamaan regresi yang telah dibentuk, pada tahap ini dilakukan perhitungan untuk memperoleh nilai prediksi jumlah publikasi *framework* pada rentang tahun 2021–2024.

## 5. Hasil Prediksi

Hasil dari proses prediksi disajikan kepada pengguna dalam bentuk tabel yang menyajikan prediksi tren jumlah publikasi masing-masing *framework*.

Setelah menjelaskan proses penerapan metode regresi linier sederhana, bagian selanjutnya akan memaparkan tahapan-tahapan penelitian, dimulai dari proses pengumpulan data publikasi ilmiah menggunakan aplikasi *Publish or Perish*. Aplikasi ini digunakan untuk

melakukan pencarian literatur secara otomatis dengan mengambil sumber dari Google Scholar. Kata kunci yang digunakan dalam pencarian disesuaikan dengan nama *framework* yang menjadi objek penelitian, yaitu “React JS”, “Laravel”, dan “Bootstrap”. Untuk setiap *framework*, data dikumpulkan dalam dua pendekatan, yaitu melalui pencarian berdasarkan judul artikel (*title*) dan berdasarkan kata kunci yang terdapat dalam artikel ilmiah (*keyword*). Tampilan halaman awal aplikasi *Publish or Perish* bisa dilihat pada Gambar 3.3 dibawah ini.



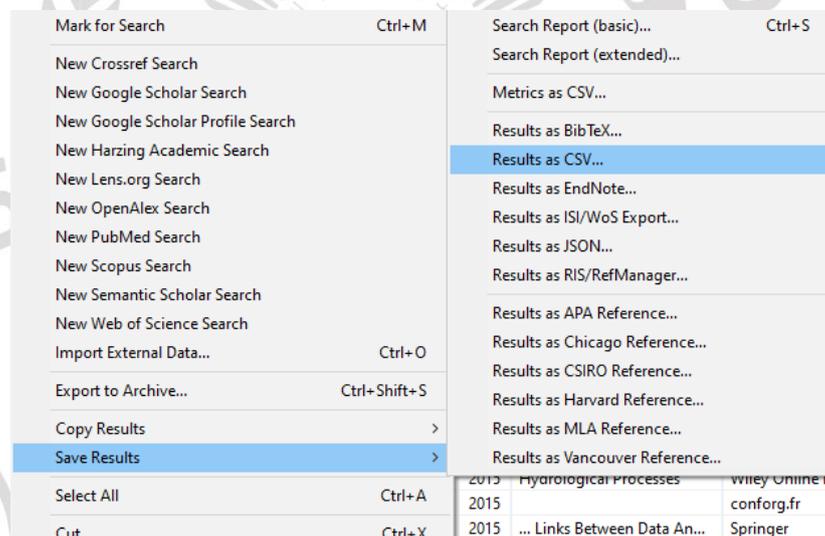
**Gambar 3. 3** Tampilan Halaman Awal Aplikasi *Publish or Perish*

Aplikasi *Publish or Perish* menyediakan berbagai pilihan sumber data, seperti Crossref, Google Scholar, Lens.org, Scopus, Semantic Scholar, Google Scholar Profile, dan lainnya. Dalam penelitian ini, data dikumpulkan melalui *web* Google Scholar, yang mencakup jurnal maupun karya ilmiah lain yang berkaitan dengan objek penelitian.



**Gambar 3. 4** Tampilan Kolom Pencarian Aplikasi *Publish or Perish*

Gambar 3.4 menunjukkan tampilan antarmuka kolom pencarian pada aplikasi *Publish or Perish* yang digunakan untuk mengumpulkan data publikasi dari Google Scholar berdasarkan kata kunci atau judul tertentu. Kolom *Title Words* digunakan untuk mencari artikel atau jurnal berdasarkan judulnya, sedangkan kolom *Keywords* digunakan untuk mencari artikel berdasarkan kata kunci yang terdapat di dalam metadata atau isi artikel, dan kolom *Years* menentukan rentang waktu publikasi artikel atau jurnal yang dicari. Dalam penelitian ini, pencarian dilakukan secara terpisah untuk setiap tahun, yaitu dari tahun 2005 hingga 2024, sehingga proses pencarian perlu diulang sebanyak 20 kali untuk masing-masing jenis pencarian (*title* dan *keyword*). Hal ini dilakukan untuk memperoleh jumlah data yang seakurat mungkin, karena setiap proses pencarian dalam aplikasi dibatasi hingga maksimal 1000 hasil. Setelah kolom-kolom yang diperlukan diisi, proses pencarian dilakukan dengan menekan tombol *Search* pada aplikasi *Publish or Perish*.



**Gambar 3.5** Format Penyimpanan Hasil Pencarian Data

Setelah proses pencarian selesai dilakukan, data kemudian disimpan dalam format CSV seperti pada Gambar 3.5. Data tersebut kemudian diubah ke format (.xlsx) untuk memudahkan proses pengolahan data yang dilakukan menggunakan Microsoft Excel. Proses pengumpulan data menghasilkan total 68.956 data mentah, adapun rincian jumlah data

mentah yang berhasil dikumpulkan menggunakan aplikasi *Publish or Perish* bisa dilihat pada Tabel 3.1.

**Tabel 3. 1** Rincian Pengumpulan Data Menggunakan Aplikasi *Publish or Perish*

<b>Framework</b>	<b>Berdasarkan Title</b>	<b>Berdasarkan Keyword</b>
Bootstrap	15279	19967
Laravel	3248	9898
React JS	588	19976

Untuk memberikan gambaran yang lebih jelas mengenai jumlah publikasi penelitian *framework* pada rentang tahun 2005 – 2024, dibuatkan rangkuman data dalam Tabel 3.2 berikut.

**Tabel 3. 2** Rincian Jumlah Publikasi Penelitian *Framework*

		<b>Bootstrap</b>	<b>Laravel</b>	<b>React JS</b>
<b>2005</b>	<b>Title</b>	520	0	0
	<b>Keyword</b>	962	6	988
<b>2006</b>	<b>Title</b>	572	0	0
	<b>Keyword</b>	957	4	987
<b>2007</b>	<b>Title</b>	548	0	0
	<b>Keyword</b>	943	13	983
<b>2008</b>	<b>Title</b>	555	2	0
	<b>Keyword</b>	956	12	988
<b>2009</b>	<b>Title</b>	543	0	0
	<b>Keyword</b>	946	15	992
<b>2010</b>	<b>Title</b>	580	0	0
	<b>Keyword</b>	963	32	996
<b>2011</b>	<b>Title</b>	573	5	0
	<b>Keyword</b>	939	37	989
<b>2012</b>	<b>Title</b>	670	8	0

	<b>Keyword</b>	948	49	993
<b>2013</b>	<b>Title</b>	722	15	1
	<b>Keyword</b>	949	103	993
<b>2014</b>	<b>Title</b>	796	32	1
	<b>Keyword</b>	943	236	993
<b>2015</b>	<b>Title</b>	862	61	6
	<b>Keyword</b>	955	416	997
<b>2016</b>	<b>Title</b>	849	87	8
	<b>Keyword</b>	934	821	997
<b>2017</b>	<b>Title</b>	879	125	19
	<b>Keyword</b>	946	922	990
<b>2018</b>	<b>Title</b>	864	161	28
	<b>Keyword</b>	954	919	990
<b>2019</b>	<b>Title</b>	873	248	38
	<b>Keyword</b>	950	945	993
<b>2020</b>	<b>Title</b>	823	273	63
	<b>Keyword</b>	955	942	984
<b>2021</b>	<b>Title</b>	812	379	77
	<b>Keyword</b>	949	966	966
<b>2022</b>	<b>Title</b>	803	473	96
	<b>Keyword</b>	958	966	977
<b>2023</b>	<b>Title</b>	751	618	120
	<b>Keyword</b>	946	972	960
<b>2024</b>	<b>Title</b>	779	653	105
	<b>Keyword</b>	949	963	970

Tabel 3.2 menyajikan rincian jumlah publikasi artikel ilmiah yang membahas *framework* React JS, Laravel, dan Bootstrap pada masing-masing tahun selama periode 2005 hingga 2024. Namun, beberapa data publikasi yang berhasil dikumpulkan tidak mencantumkan informasi tahun terbit secara lengkap, sehingga tidak dimasukkan ke dalam tabel. Data

tersebut tidak digunakan dalam proses analisis karena tidak dapat diklasifikasikan ke dalam rentang waktu penelitian, yaitu tahun 2005 hingga 2024.

Setelah data dikumpulkan, langkah selanjutnya adalah melakukan *preprocessing* atau pra-pemrosesan data. Tahap ini bertujuan untuk memastikan bahwa data yang akan digunakan dalam proses analisis dan prediksi benar-benar relevan dan bersih dari elemen-elemen yang tidak diperlukan. Proses *preprocessing* dilakukan terhadap data yang telah diekspor dari aplikasi *Publish or Perish* dalam format CSV dan kemudian dikonversi ke format Excel (.xlsx) agar lebih mudah dikelola.

Langkah pertama dalam *preprocessing* adalah menggabungkan data hasil pencarian berdasarkan judul (*title*) dan kata kunci (*keywords*) untuk masing-masing *framework*. Penggabungan ini dilakukan untuk memperluas cakupan data dan mengidentifikasi artikel atau jurnal yang mungkin hanya terdeteksi oleh salah satu jenis pencarian. Setelah data digabungkan, tahap selanjutnya adalah menghapus data duplikat atau entri yang sama agar tidak terjadi pengulangan data dalam proses analisis dan prediksi.

Selanjutnya, dilakukan penyaringan data penelitian di Indonesia untuk memastikan bahwa data yang digunakan hanya mencakup publikasi yang relevan dengan konteks penelitian di Indonesia. Penyaringan ini dilakukan dengan cara membaca dan memeriksa judul artikel, nama penulis, nama jurnal, dan *publisher*. Selain itu, jika afiliasi peneliti atau lokasi penelitian menunjukkan bahwa penelitian tersebut dilakukan di Indonesia, maka artikel tersebut akan dianggap relevan dan dimasukkan ke dalam dataset. Untuk membantu proses penyaringan data berjalan lebih efisien dan sistematis, peneliti memanfaatkan bahasa pemrograman Python dengan dukungan *library* Streamlit dan Pandas. Dengan pendekatan ini, data dapat difilter berdasarkan indikator-indikator yang sudah dijelaskan sebelumnya, sehingga proses penyaringan menjadi lebih terstruktur dan akurat.

Setelah proses penyaringan selesai, diperoleh dataset baru yang sudah bersih dan siap untuk digunakan. Dataset tersebut akan digunakan dalam proses penerapan metode regresi linier sederhana untuk memodelkan tren penggunaan *framework* dari waktu ke waktu.

#### 1. Penggabungan Data

Penggabungan data dilakukan dengan menyatukan hasil pencarian berdasarkan *title* dan *keyword* untuk setiap *framework* ke dalam satu lembar kerja Excel. Proses ini dilakukan dengan menyalin seluruh data dari masing-masing hasil pencarian, kemudian menggabungkannya secara manual ke dalam satu *worksheet* agar data lebih mudah dikelola dan dianalisis pada tahap selanjutnya. Hasil penggabungan data berdasarkan pencarian judul dan kata kunci untuk masing-masing *framework* setiap tahunnya akan ditampilkan pada Tabel 3.3.

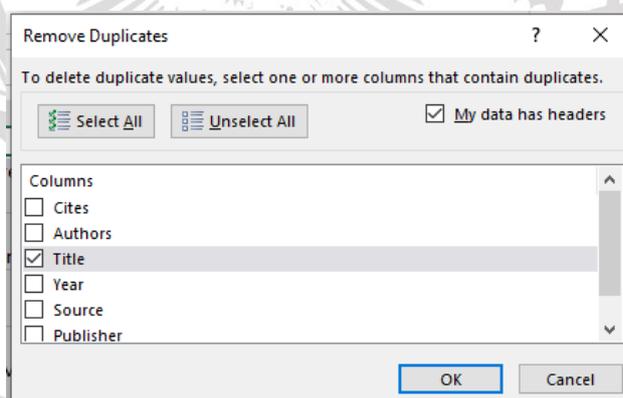
**Tabel 3. 3** Hasil Penggabungan Data *Title* dan *Keyword*

Tahun	Framework		
	Bootstrap	Laravel	React JS
2005	1482	6	988
2006	1529	4	987
2007	1491	13	983
2008	1511	14	988
2009	1489	15	992
2010	1543	32	996
2011	1512	42	989
2012	1618	57	993
2013	1671	118	994
2014	1739	268	994
2015	1817	477	1003
2016	1783	908	1005
2017	1825	1047	1009

<b>2018</b>	1818	1080	1018
<b>2019</b>	1823	1193	1031
<b>2020</b>	1778	1215	1047
<b>2021</b>	1761	1345	1043
<b>2022</b>	1761	1439	1073
<b>2023</b>	1697	1590	1080
<b>2024</b>	1728	1616	1075

## 2. Penghapusan Duplikat

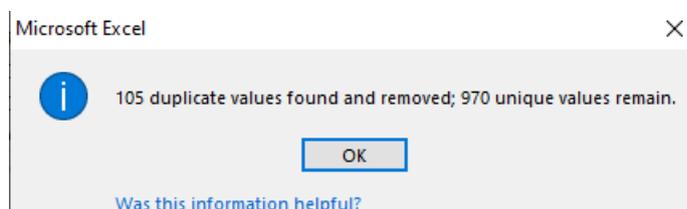
Duplikasi data dapat terjadi karena adanya artikel yang muncul baik dalam hasil pencarian berdasarkan judul maupun berdasarkan kata kunci, sehingga perlu diidentifikasi dan dihapus agar tidak terjadi pengulangan data. Proses penghapusan ini dilakukan menggunakan fitur *Remove Duplicates* yang tersedia pada menu Data di Microsoft Excel. Dalam penelitian ini, penghapusan dilakukan dengan menerapkan fitur tersebut secara khusus pada kolom *Title*. Dengan cara ini, hanya artikel dengan judul yang unik saja yang digunakan dalam proses analisis tren dan prediksi pada tahap selanjutnya.



**Gambar 3. 6** Pemilihan Kolom *Title* pada Fitur *Remove Duplicates*

Gambar 3.6 menunjukkan tampilan jendela fitur *Remove Duplicates*, pada proses ini, hanya kolom *Title* yang dipilih karena duplikasi data umumnya berasal dari artikel yang memiliki judul yang

sama. Setelah kolom ditentukan, penghapusan dilakukan dengan menekan tombol “OK”.



**Gambar 3. 7** Notifikasi Hasil *Remove Duplicates*

Setelah dilakukan penghapusan duplikat akan muncul notifikasi seperti pada Gambar 3.7 yang memuat informasi jumlah data yang terhapus dan data unik yang tersisa. Untuk mengetahui rincian jumlah data setelah penghapusan duplikat, dapat dilihat pada Tabel 3.4 berikut ini.

**Tabel 3. 4** Hasil Penghapusan Data Duplikat

Tahun	Framework		
	Bootstrap	Laravel	React JS
2005	949	6	987
2006	940	4	987
2007	927	13	982
2008	926	12	987
2009	926	15	990
2010	952	30	996
2011	926	37	989
2012	933	49	991
2013	934	102	993
2014	917	232	991
2015	931	411	996
2016	899	807	996
2017	925	914	986
2018	931	911	979
2019	937	927	991

<b>2020</b>	936	929	984
<b>2021</b>	939	959	966
<b>2022</b>	945	963	975
<b>2023</b>	946	970	967
<b>2024</b>	962	961	970

### 3. Menyaring Data Penelitian Indonesia

Penyaringan data penelitian Indonesia dilakukan untuk memastikan bahwa hanya publikasi yang benar-benar berasal dari penelitian di Indonesia yang dianalisis dalam penelitian ini. Tujuan dari proses ini adalah untuk menyaring artikel yang tidak sesuai dengan ruang lingkup penelitian, yaitu penelitian yang tidak dilakukan oleh peneliti Indonesia atau tidak diterbitkan oleh institusi di Indonesia atau juga tidak mengambil lokasi penelitian di Indonesia. Karena data yang diperoleh dari *Publish or Perish* mencakup publikasi dari berbagai negara, maka penyaringan perlu dilakukan dengan teliti. Untuk mengidentifikasi data yang tidak relevan, dilakukan analisis pada beberapa atribut yang tersedia dalam data hasil ekspor dari aplikasi *Publish or Perish*.

Salah satu indikator yang digunakan dalam proses ini adalah mengamati bahasa yang digunakan dalam judul penelitian. Penelitian yang memiliki judul dalam bahasa Indonesia dianggap sebagai representasi dari penelitian yang dilakukan di Indonesia, karena peneliti yang berasal dari Indonesia umumnya menulis dalam bahasa nasional untuk kebutuhan yang terkait dengan publikasi. Selain dari judul, proses penyaringan juga dilakukan melalui kolom *publisher*, yaitu dengan mengidentifikasi domain situs penerbit. *Publisher* yang memiliki domain .id dipastikan merupakan bagian dari institusi di Indonesia. Untuk mengetahui *publisher* yang merupakan bagian dari institusi di Indonesia dapat dilihat pada Tabel 3.5.

**Tabel 3. 5** Contoh *Publisher* Dengan Domain Indonesia

No	<i>Publisher</i>	No	<i>Publisher</i>
1	repository.unj.ac.id	11	jurnal.stis.ac.id
2	unisbank.ac.id	12	jurnal.polinema.ac.id
3	scholar.unand.ac.id	13	repository.ub.ac.id
4	jurnal.unimus.ac.id	14	jurnalteknik.unisla.ac.id
5	journal.uinsgd.ac.id	15	repository.unair.ac.id
6	digilib.uns.ac.id	16	library.stikom-bali.ac.id
7	repository.its.ac.id	17	....telkomuniversity.ac.id
8	ejournal.upnjatim.ac.id	18	jurnal.polgan.ac.id
9	ejournal3.undip.ac.id	19	digilib.unila.ac.id
10	journal.unnes.ac.id	20	jurnal.stmik-mi.ac.id

Selain domain Indonesia, ada juga beberapa platform *publisher* lain yang sering memuat penelitian dari Indonesia seperti neliti.com, core.ac.uk, academia.edu, dan ieeexplore.ieee.org. Meskipun tidak memiliki domain .id, platform-platform ini tetap dianggap relevan karena banyak digunakan oleh peneliti Indonesia untuk mendistribusikan karya ilmiahnya. Contoh dari *publisher* yang termasuk dalam kategori ini dapat dilihat pada Tabel 3.6.

**Tabel 3. 6** Contoh *Publisher* Lain Yang Memuat Penelitian Indonesia

No	<i>Publisher</i>	No	<i>Publisher</i>
1	core.ac.uk	11	jatit.org
2	Springer	12	pubs.aip.org
3	academia.edu	13	books.google.com
4	iopscience.iop.org	14	scholar.archive.org
5	researchgate.net	15	journal.binainternusa.org
6	ieeexplore.ieee.org	16	hostjournals.com
7	neliti.com	17	sistemasi.org
8	pdfs.semanticscholar.org	18	joiv.org

9	osf.io	19	journal-computing.org
10	Elsevier	20	journal-isi.org

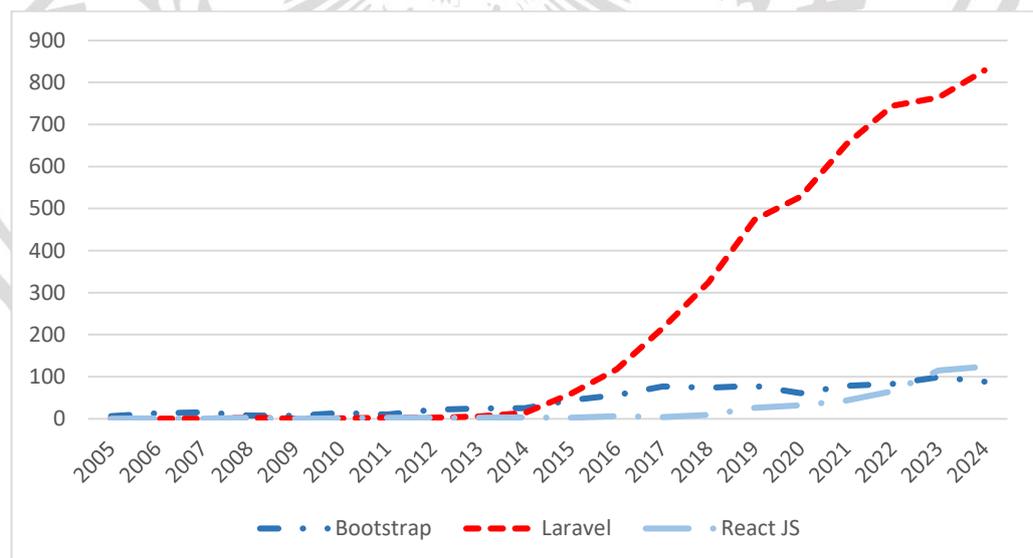
Selain judul dan *publisher*, kolom penulis atau *authors* juga digunakan sebagai acuan dalam proses penyaringan. Nama-nama penulis yang umum digunakan di Indonesia seperti Supriyanto, Hanifah, Nugroho, Udin, Hartono, dan lainnya menjadi indikator tambahan bahwa penelitian tersebut kemungkinan besar berasal dari Indonesia. Proses penyaringan dilakukan dengan menggunakan fungsi Streamlit dan Pandas dari *library* Python berdasarkan aspek-aspek tersebut, data yang tersisa diperiksa kembali secara manual untuk memastikan bahwa tidak ada artikel yang tidak relevan masih tercampur di dalam dataset akhir. Pemeriksaan manual ini dilakukan untuk meningkatkan akurasi data yang akan digunakan dalam tahap analisis dan prediksi. Hasil akhir dari proses penghapusan data yang tidak relevan disajikan pada Tabel 3.7.

**Tabel 3. 7** Hasil Penghapusan Data Yang Tidak Relevan

Tahun	Framework		
	Bootstrap	Laravel	React JS
2005	6	0	0
2006	13	0	1
2007	15	0	0
2008	8	2	2
2009	7	0	0
2010	14	1	1
2011	10	2	2
2012	21	2	2
2013	24	5	2
2014	25	14	3
2015	44	59	2

<b>2016</b>	56	118	6
<b>2017</b>	77	216	4
<b>2018</b>	74	324	9
<b>2019</b>	78	473	26
<b>2020</b>	61	528	32
<b>2021</b>	78	653	43
<b>2022</b>	83	744	65
<b>2023</b>	99	764	115
<b>2024</b>	88	829	124

Data yang disajikan pada Tabel 3.7 merupakan hasil akhir setelah dilakukan penghapusan duplikat dan penyaringan berdasarkan judul, *publisher*, serta nama penulis yang mengindikasikan bahwa artikel tersebut berasal dari Indonesia. Tabel ini menggambarkan tren peningkatan publikasi *framework* dari tahun ke tahun, terutama untuk Laravel dan React JS, yang menunjukkan pertumbuhan signifikan dalam dekade terakhir. Dari Tabel 3.7 dibuatkan ilustrasi berupa grafik sebagai berikut :



**Gambar 3. 8** Grafik Jumlah Publikasi *Framework*

Gambar 3.8 menunjukkan tren jumlah publikasi artikel untuk masing-masing *framework*, yaitu Bootstrap, Laravel, dan React JS, selama periode tahun 2005 hingga 2024. Berdasarkan grafik, terlihat bahwa Laravel mengalami peningkatan yang sangat tajam, terutama sejak tahun 2015. Jumlah publikasinya terus meningkat secara konsisten setiap tahun hingga menjadi *framework* dengan jumlah publikasi tertinggi pada tahun 2024. Sementara itu, Bootstrap menunjukkan tren yang stabil dan meningkat secara perlahan. Peningkatannya tampak mulai terlihat lebih jelas setelah tahun 2013, namun tetap berada di bawah angka publikasi Laravel. Adapun React JS memiliki jumlah publikasi yang paling rendah di antara ketiga *framework*, dengan pertumbuhan yang baru mulai tampak signifikan sejak sekitar tahun 2018, dan terus meningkat secara bertahap hingga tahun 2024.

### 3.3 Representasi Model

Setelah menyelesaikan proses pengumpulan dan pemrosesan data, tahap selanjutnya adalah menerapkan metode Regresi Linier Sederhana untuk membentuk model prediksi tren. Model ini digunakan untuk menganalisis hubungan antara tahun sebagai variabel independen (X) dan jumlah publikasi *framework* sebagai variabel dependen (Y), dengan tujuan untuk mengetahui tren perkembangan tiap *framework* dalam penelitian di Indonesia. Penerapan metode ini dilakukan dengan menghitung parameter-parameter model seperti koefisien regresi (b) dan konstanta (a), yang kemudian digunakan untuk membentuk persamaan regresi linier. Proses pembentukan model dilakukan menggunakan data latih (training data), yaitu data jumlah publikasi dari tahun 2005 hingga 2020. Setelah model terbentuk, dilakukan proses prediksi terhadap data uji (testing data) yaitu tahun 2021 hingga 2024 dengan memasukkan nilai tahun ke dalam model persamaan yang telah dibentuk. Berikut adalah proses perhitungan menggunakan metode Regresi Linier Sederhana.

### 1. Menentukan Variabel Bebas dan Terikat

Variabel bebas (X) diambil dari nilai tahun 2005–2020, yang kemudian dikonversi menjadi angka 1–16 untuk mempermudah perhitungan. Variabel terikat (Y) adalah jumlah publikasi untuk masing-masing *framework* pada setiap tahun yang telah disajikan pada Tabel 3.7. Dengan demikian, untuk setiap *framework* akan didapatkan sepasang data X dan Y yang digunakan dalam perhitungan regresi.

### 2. Model Persamaan Regresi Linier

Pada tahap ini, dilakukan perhitungan untuk menentukan model regresi linier sederhana sesuai dengan persamaan (2.1). Untuk membentuk model persamaan regresi, dilakukan perhitungan nilai koefisien regresi (b) dan konstanta (a) menggunakan data latih yang telah tersedia. Untuk menghitung koefisien (b) dan konstanta (a) dapat menggunakan persamaan berikut :

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2} \quad (3.1)$$

$$a = \frac{(\sum Y) - b(\sum X)}{n} \quad (3.2)$$

Dengan keterangan :

X adalah tahun (dikonversi menjadi angka 1–16).

Y adalah jumlah publikasi *framework*.

n adalah jumlah data.

#### a. Model Persamaan Bootstrap

Model regresi linier sederhana untuk Bootstrap dibentuk menggunakan data latih berupa jumlah publikasi *framework* Bootstrap pada tahun 2005–2020, dengan tahun sebagai variabel X dan jumlah publikasi sebagai Y. Langkah pertama perhitungannya adalah menghitung  $\sum X$ ,  $\sum Y$ ,  $\sum XY$ , dan  $\sum X^2$ . Nilai-nilai tersebut dapat dilihat pada Tabel 3.8 berikut.

**Tabel 3. 8** Perhitungan Data *Framework* Bootstrap

<b>Bootstrap</b>
------------------

	X (tahun)	Y (publikasi)	XY	X <sup>2</sup>
	1	6	6	1
	2	13	26	4
	3	15	45	9
	4	8	32	16
	5	7	35	25
	6	14	84	36
	7	10	70	49
	8	21	168	64
	9	24	216	81
	10	25	250	100
	11	44	484	121
	12	56	672	144
	13	77	1001	169
	14	74	1036	196
	15	78	1170	225
	16	61	976	256
<b>Jumlah</b>	<b>136</b>	<b>533</b>	<b>6271</b>	<b>1496</b>

Langkah berikutnya adalah menghitung koefisien (b) menggunakan persamaan (3.1).

$$b = \frac{16(6271) - (136)(533)}{16(1496) - (136)^2}$$

$$b = \frac{100336 - 72488}{23936 - 18496} = \frac{27848}{5440} = 5,119$$

Setelah menemukan hasil dari koefisien (b), selanjutnya adalah menghitung konstanta (a) menggunakan persamaan (3.2).

$$a = \frac{533 - (5,119)(136)}{16}$$

$$a = \frac{533 - 696,184}{16} = \frac{-163,184}{16} = -10,2$$

Setelah mendapatkan nilai a dan b, maka model persamaan regresi untuk Bootstrap adalah :

$$Y = -10,2 + 5,119(X)$$

b. Model Persamaan Laravel

Model regresi linier sederhana untuk Laravel dibentuk menggunakan data latih berupa jumlah publikasi *framework* Laravel pada tahun 2005–2020, dengan tahun sebagai variabel X dan jumlah publikasi sebagai Y. Langkah pertama perhitungannya adalah menghitung  $\sum X$ ,  $\sum Y$ ,  $\sum XY$ , dan  $\sum X^2$ . Nilai-nilai tersebut dapat dilihat pada Tabel 3.9 berikut.

**Tabel 3. 9** Perhitungan Data *Framework* Laravel

<b>Laravel</b>				
	X (tahun)	Y (publikasi)	XY	X <sup>2</sup>
	1	0	0	1
	2	0	0	4
	3	0	0	9
	4	2	8	16
	5	0	0	25
	6	1	6	36
	7	2	14	49
	8	2	16	64
	9	5	45	81
	10	14	140	100
	11	59	649	121
	12	118	1416	144
	13	216	2808	169
	14	324	4536	196
	15	473	7095	225
	16	528	8448	256
<b>Jumlah</b>	<b>136</b>	<b>1744</b>	<b>25181</b>	<b>1496</b>

Langkah berikutnya adalah menghitung koefisien (b) menggunakan persamaan (3.1).

$$b = \frac{16(25181) - (136)(1744)}{16(1496) - (136)^2}$$

$$b = \frac{402896 - 237184}{23936 - 18496} = \frac{165712}{5440} = 30,462$$

Setelah menemukan hasil dari koefisien (b), selanjutnya adalah menghitung konstanta (a) menggunakan persamaan (3.2).

$$a = \frac{1744 - (30,462)(136)}{16}$$

$$a = \frac{1744 - 4142,832}{16} = \frac{-2398,832}{16} = -149,927$$

Setelah mendapatkan nilai a dan b, maka model persamaan regresi untuk Laravel adalah :

$$Y = -149,927 + 30,462(X)$$

c. Model Persamaan React JS

Model regresi linier sederhana untuk React JS dibentuk menggunakan data latih berupa jumlah publikasi *framework* React JS pada tahun 2005–2020, dengan tahun sebagai variabel X dan jumlah publikasi sebagai Y. Langkah pertama perhitungannya adalah menghitung  $\sum X$ ,  $\sum Y$ ,  $\sum XY$ , dan  $\sum X^2$ . Nilai-nilai tersebut dapat dilihat pada Tabel 3.10 berikut.

**Tabel 3. 10** Perhitungan Data *Framework* React JS

React JS				
	X (tahun)	Y (publikasi)	XY	X <sup>2</sup>
	1	0	0	1
	2	1	2	4
	3	0	0	9
	4	2	8	16
	5	0	0	25
	6	1	6	36

	7	2	14	49
	8	2	16	64
	9	2	18	81
	10	3	30	100
	11	2	22	121
	12	6	72	144
	13	4	52	169
	14	9	126	196
	15	26	390	225
	16	32	512	256
<b>Jumlah</b>	<b>136</b>	<b>92</b>	<b>1268</b>	<b>1496</b>

Langkah berikutnya adalah menghitung koefisien (b) menggunakan persamaan (3.1).

$$b = \frac{16(1268) - (136)(92)}{16(1496) - (136)^2}$$

$$b = \frac{20288 - 12512}{23936 - 18496} = \frac{7776}{5440} = 1,429$$

Setelah menemukan hasil dari koefisien (b), selanjutnya adalah menghitung konstanta (a) menggunakan persamaan (3.2).

$$a = \frac{92 - (1,429)(136)}{16}$$

$$a = \frac{92 - 194,344}{16} = \frac{-102,344}{16} = -6,396$$

Setelah mendapatkan nilai a dan b, maka model persamaan regresi untuk React JS adalah :

$$Y = -6,396 + 1,429(X)$$

### 3. Melakukan Prediksi

Dengan model persamaan regresi yang telah terbentuk, proses prediksi dapat dilakukan dengan memasukkan nilai X (tahun) ke dalam model persamaan. Sebagai contoh perhitungan, digunakan tahun 2021.

Hasil dari perhitungan ini akan menunjukkan jumlah publikasi yang diprediksi pada tahun 2021 untuk masing-masing *framework*.

a. Prediksi *Framework* Bootstrap

Persamaan regresi linier sederhana yang digunakan untuk *framework* Bootstrap adalah sebagai berikut :

$$Y = -10,2 + 5,119(X)$$

Persamaan ini digunakan untuk memprediksi tren publikasi pada tahun 2021. Dalam hal ini, nilai X menunjukkan urutan tahun sejak 2005, sehingga untuk tahun 2021,  $X = 17$ . Nilai tersebut dimasukkan ke dalam persamaan untuk menghitung prediksi tren publikasi *framework* Bootstrap pada tahun 2021. Berikut adalah perhitungan prediksi untuk tren publikasi *framework* Bootstrap pada tahun 2021.

$$Y = -10,2 + 5,119(17)$$

$$Y = -10,2 + 87,023 = 76,823$$

Dengan demikian, hasil prediksi menunjukkan bahwa tren publikasi *framework* Bootstrap pada tahun 2021 adalah **77 penelitian**.

b. Prediksi *Framework* Laravel

Persamaan regresi linier sederhana yang digunakan untuk *framework* Laravel adalah sebagai berikut :

$$Y = -149,927 + 30,462(X)$$

Persamaan ini digunakan untuk memprediksi tren publikasi pada tahun 2021. Dalam hal ini, nilai X menunjukkan urutan tahun sejak 2005, sehingga untuk tahun 2021,  $X = 17$ . Nilai tersebut dimasukkan ke dalam persamaan untuk menghitung prediksi tren publikasi *framework* Laravel pada tahun 2021. Berikut adalah perhitungan prediksi untuk tren publikasi *framework* Laravel pada tahun 2021.

$$Y = -149,927 + 30,462(17)$$

$$Y = -149,927 + 517,854 = 367,927$$

Dengan demikian, hasil prediksi menunjukkan bahwa tren publikasi *framework* Laravel pada tahun 2021 adalah **368 penelitian**.

c. Prediksi *Framework* React JS

Persamaan regresi linier sederhana yang digunakan untuk *framework* React JS adalah sebagai berikut :

$$Y = -6,396 + 1,429(X)$$

Persamaan ini digunakan untuk memprediksi tren publikasi pada tahun 2021. Dalam hal ini, nilai X menunjukkan urutan tahun sejak 2005, sehingga untuk tahun 2021,  $X = 17$ . Nilai tersebut dimasukkan ke dalam persamaan untuk menghitung prediksi tren publikasi *framework* React JS pada tahun 2021. Berikut adalah perhitungan prediksi untuk tren publikasi *framework* React JS pada tahun 2021.

$$Y = -6,396 + 1,429(17)$$

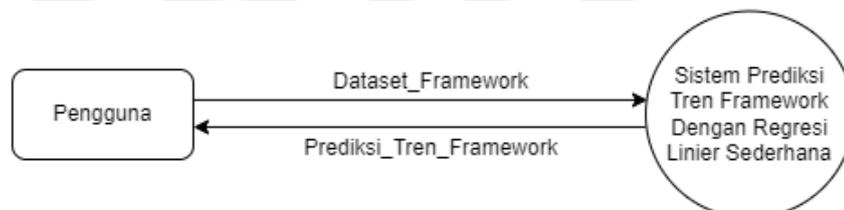
$$Y = -6,396 + 24,293 = 17,897$$

Dengan demikian, hasil prediksi menunjukkan bahwa tren publikasi *framework* React JS pada tahun 2021 adalah **18 penelitian**.

### 3.4 Perancangan Sistem

#### 3.4.1 Diagram Konteks

Diagram konteks merupakan diagram sederhana dalam pemodelan sistem yang menggambarkan interaksi antara sistem utama dengan entitas eksternal seperti pengguna, atau pihak luar melalui alur data.

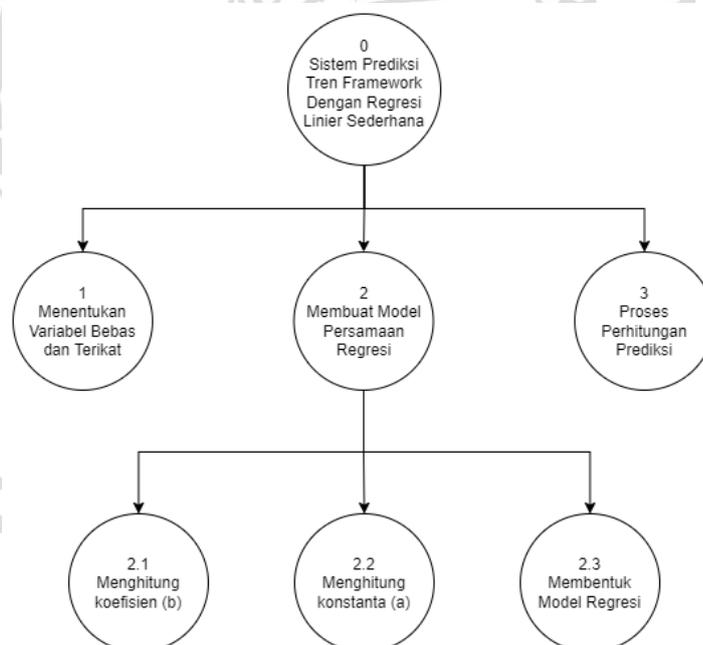


**Gambar 3. 9** Diagram Konteks Sistem

Diagram konteks ini menjelaskan hubungan antara pengguna dan sistem yang digunakan untuk melakukan prediksi tren penggunaan *framework* dengan metode Regresi Linier Sederhana. Pengguna menyediakan masukan berupa *Dataset\_Framework*, yaitu data yang telah diproses dan berisi informasi jumlah publikasi *framework* selama periode waktu tertentu. Selanjutnya, sistem akan mengolah data tersebut melalui proses analisis tren dan perhitungan model regresi guna menghasilkan *Prediksi\_Tren\_Framework*. Hasil ini kemudian dikirimkan kembali kepada pengguna sebagai keluaran sistem.

### 3.4.2 Diagram Jenjang

Diagram berjenjang menunjukkan hirarki dari proses-proses utama yang terjadi di dalam sistem. Dalam pengembangan sistem prediksi tren *framework website*, diagram berjenjang digunakan untuk mempermudah visualisasi alur kerja sistem sebelum dijabarkan ke dalam *Data Flow Diagram* (DFD) yang lebih rinci.



**Gambar 3. 10** Diagram Jenjang Sistem

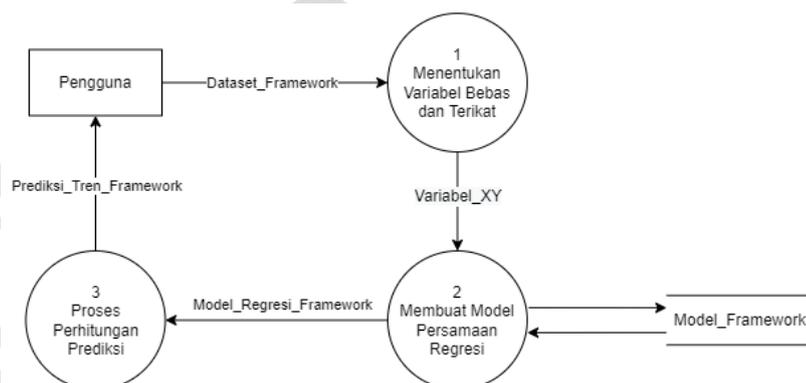
Gambar 3.10 menggambarkan diagram berjenjang dari sistem dengan urutan yang terdiri dari :

1. Top Level : Sistem Prediksi Tren *Framework* dengan menggunakan Regresi Linier Sederhana, yang berperan sebagai pusat proses analisis tren dan prediksi jumlah publikasi.
2. Level 1 : Diagram ini menunjukkan tahapan utama sistem yang melibatkan proses perhitungan regresi linier sederhana.
  - a. Menentukan Variabel Bebas dan Terikat: Menentukan variabel bebas (X) dan variabel terikat (Y) dari dataset publikasi yang tersedia.
  - b. Membuat Model Persamaan Regresi: Menghitung koefisien (b) dan konstanta (a) untuk membentuk model regresi masing-masing *framework* berdasarkan data latih.
  - c. Proses Perhitungan Prediksi: Menggunakan model regresi yang sudah dibentuk untuk menghitung nilai prediksi jumlah publikasi masing-masing *framework*.
3. Level 2 : Diagram ini menggambarkan rincian proses yang pada level sebelumnya.
  - a. Menghitung Koefisien (b): Menentukan nilai gradien garis regresi, yaitu besarnya perubahan variabel Y terhadap setiap kenaikan unit variabel X.
  - b. Menghitung Konstanta (a): Menghitung titik potong garis regresi terhadap sumbu Y.
  - c. Membentuk Model Regresi: Menggabungkan nilai koefisien dan konstanta untuk membentuk model regresi masing-masing *framework*.

### 3.4.3 Data Flow Diagram

*Data Flow Diagram* (DFD) sistem prediksi tren *framework* ini berfungsi untuk memperlihatkan hubungan antara aliran data, entitas pengguna, serta tahapan proses yang terjadi di dalam sistem. DFD ini menyusun langkah-langkah mulai dari penentuan variabel

bebas dan terikat hingga penyajian nilai hasil prediksi. Setiap proses dalam diagram ini menunjukkan bagaimana data diproses secara bertahap untuk menghasilkan informasi prediksi yang berguna bagi pengguna. Gambar 3.11 berikut menyajikan DFD Level 1 yang menggambarkan alur data dan proses utama dalam sistem.

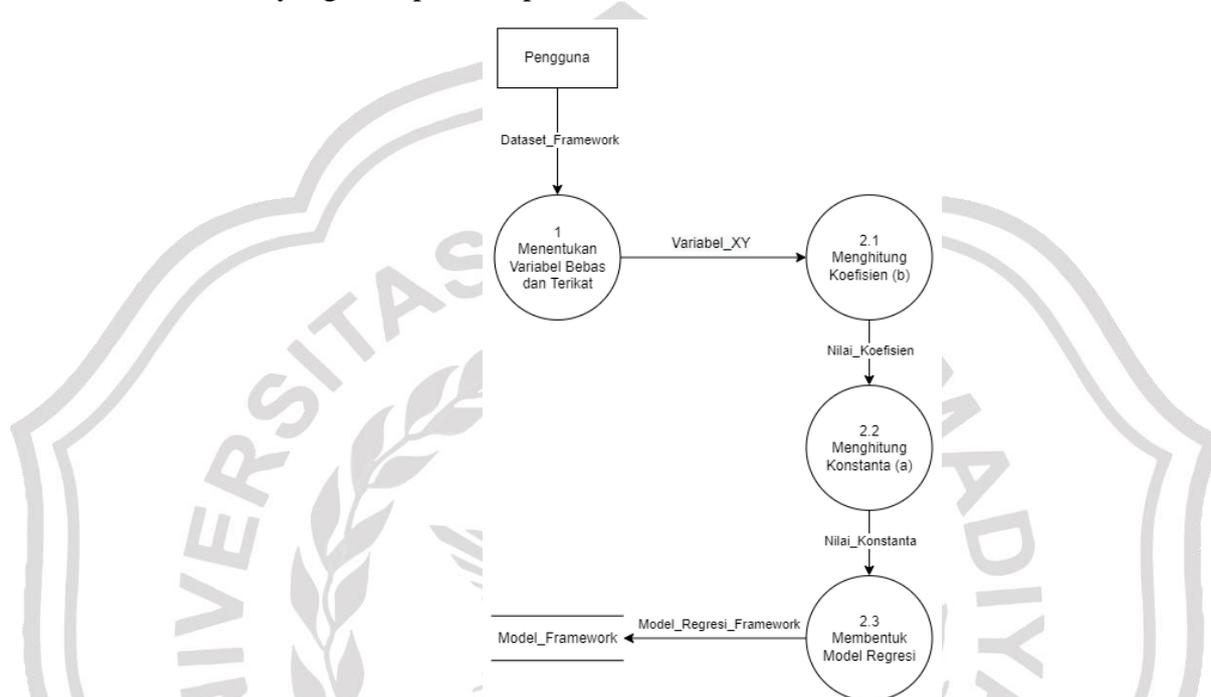


**Gambar 3. 11** DFD Level 1 Sistem

Pada Gambar 3.11, ditunjukkan alur sistem prediksi tren *framework* yang terdiri dari tiga proses utama. Proses 1.0 adalah Menentukan Variabel Bebas dan Terikat, yaitu proses di mana sistem menerima *Dataset\_Framework* dari pengguna dan mengidentifikasi variabel bebas (tahun) dan variabel terikat (jumlah publikasi) sebagai dasar perhitungan regresi. Proses 2.0 adalah Membuat Model Persamaan Regresi, yang menggunakan data variabel untuk menghitung nilai koefisien (b) dan konstanta (a), kemudian membentuk *Model\_Regresi\_Framework* dan menyimpannya kedalam Data Store *Model\_Framework*. Terakhir, Proses 3.0 yaitu Proses Perhitungan Prediksi, di mana sistem menggunakan model regresi yang telah dibuat untuk menghitung prediksi jumlah publikasi masing-masing *framework* dan menghasilkan output *Prediksi\_Tren\_Framework* yang akan disampaikan kepada pengguna.

Proses 2.0 Membuat Model Persamaan Regresi memiliki beberapa tahapan perhitungan yang cukup penting dan berurutan,

seperti perhitungan nilai koefisien dan konstanta regresi. Oleh karena itu, diperlukan pemodelan lebih rinci melalui DFD Level 2 untuk menggambarkan aliran data pada proses ini secara lebih spesifik. Alur detail dari proses ini disajikan dalam Gambar 3.12 yang merupakan representasi DFD Level 2 untuk Proses 2.0.



**Gambar 3. 12** DFD Level 2 Proses 2.0 Membuat Model Persamaan Regresi

Gambar 3.12 merupakan DFD Level 2 dari proses 2.0 Membuat Model Persamaan Regresi. Diagram ini menjabarkan lebih lanjut proses yang dilakukan sistem untuk membentuk model regresi linier sederhana dari data yang telah diinputkan sebelumnya (Variabel\_XY). Proses ini dilakukan dalam tiga langkah utama, dimulai dari perhitungan koefisien (b), perhitungan konstanta (a), hingga pembentukan model regresi untuk masing-masing *framework*. Penjelasan setiap langkahnya adalah sebagai berikut:

1. Proses 2.1 Menghitung Koefisien (b)

Sistem menghitung nilai koefisien regresi (b) berdasarkan rumus regresi linier sederhana menggunakan input

Variabel\_XY. Nilai koefisien (b) akan menunjukkan perubahan jumlah publikasi setiap tahunnya.

## 2. Proses 2.2 Menghitung Konstanta (a)

Setelah nilai koefisien diperoleh, sistem menghitung nilai konstanta (a), yaitu titik potong garis regresi pada sumbu Y ketika X bernilai nol.

## 3. Proses 2.3 Membentuk Model Regresi

Berdasarkan nilai koefisien dan konstanta yang telah dihitung, sistem membentuk model persamaan regresi untuk masing-masing *framework*. Output dari proses ini adalah Model\_Regresi\_Framework yang akan disimpan ke dalam Data Store Model\_Framework dan digunakan dalam proses prediksi selanjutnya.

### 3.5 Perancangan Basis Data

Perancangan basis data bertujuan untuk mengatur penyimpanan data secara sistematis agar mendukung kebutuhan sistem dalam melakukan analisis dan prediksi tren *framework*. Untuk memudahkan pengelolaan dan integrasi data antar proses, data dibagi ke dalam beberapa tabel utama. Terdapat empat tabel utama yang digunakan, yaitu tabel *user*, tabel *framework*, tabel dataset, dan tabel model. Berikut adalah rincian penjelasan dari masing-masing tabel tersebut.

#### 3.5.1 Tabel User

Tabel ini berfungsi untuk menyimpan data akun pengguna yang memiliki akses ke sistem. Rincian struktur tabel disajikan pada

Tabel 3.11 :

**Tabel 3.11** Tabel User

Nama	Tipe Data	Deskripsi
id_user	Integer	PK
username	Varchar	
password	Varchar	

### 3.5.2 Tabel Framework

Tabel ini berfungsi untuk menyimpan daftar *framework* yang menjadi objek analisis dalam penelitian. Tabel ini terhubung dengan tabel User melalui *foreign key*. Rincian struktur tabel disajikan pada Tabel 3.12 :

**Tabel 3. 12** Tabel Framework

Nama	Tipe Data	Deskripsi
id_framework	Integer	PK
id_user	Integer	FK
nama_framework	Varchar	

### 3.5.3 Tabel Dataset Framework

Tabel ini berfungsi untuk menyimpan data jumlah publikasi berdasarkan *framework* dan tahun. Tabel ini terhubung dengan tabel Framework melalui *foreign key*. Rincian struktur tabel disajikan pada Tabel 3.13 :

**Tabel 3. 13** Tabel Dataset Framework

Nama	Tipe Data	Deskripsi
id_publikasi	Integer	PK
id_framework	Integer	FK
tahun	Integer	
jumlah	Integer	

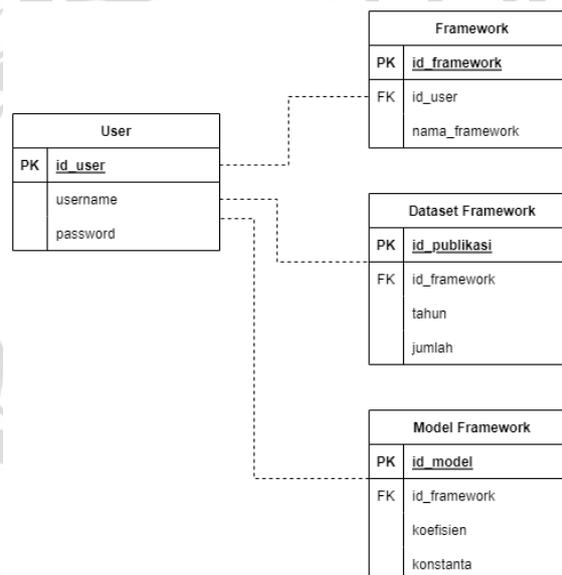
### 3.5.4 Tabel Model Framework

Tabel ini digunakan untuk menyimpan model regresi masing-masing *framework*. Tabel ini juga terhubung dengan tabel Framework melalui *foreign key*. Adapun struktur tabelnya dapat dilihat pada Tabel 3.14 :

**Tabel 3. 14** Tabel Model Framework

Nama	Tipe Data	Deskripsi
id_model	Integer	PK
id_framework	Integer	FK
koefisien	Double	
konstanta	Double	

Database pada sistem ini dirancang untuk mengelola data-data yang digunakan dalam proses analisis tren *framework*. Untuk memperjelas hubungan antar tabel, struktur basis data ditampilkan dalam bentuk *Entity Relationship Diagram* (ERD) yang ditunjukkan pada Gambar 3.13.

**Gambar 3. 13** *Entity Relationship Diagram* (ERD)

Pada Gambar 3.13 ditampilkan ERD dari sistem prediksi tren penggunaan *framework*. Tabel User digunakan untuk menyimpan informasi akun pengguna, seperti *username* dan *password* untuk keperluan autentikasi ke dalam sistem. Tabel Framework digunakan untuk mencatat nama-nama *framework* yang dianalisis, serta mengaitkannya dengan pengguna yang menambahkannya melalui atribut *id\_user* sebagai *foreign key*. Tabel

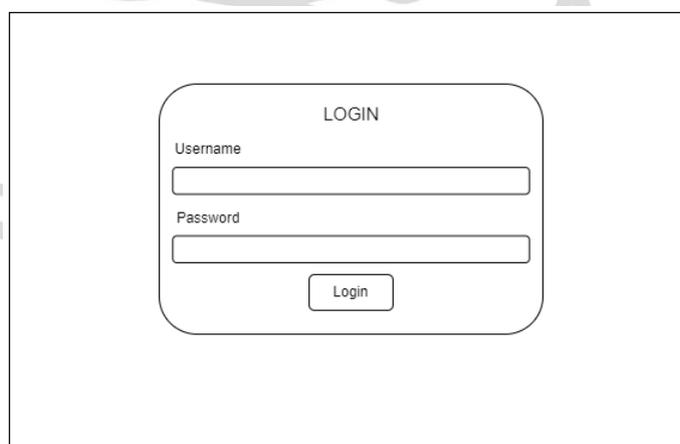
Dataset Framework berfungsi menyimpan data jumlah publikasi tiap *framework* berdasarkan tahun, yang menjadi dasar dalam proses analisis dan prediksi tren penggunaan *framework*. Sedangkan Tabel Model Framework menyimpan hasil perhitungan regresi dalam bentuk nilai koefisien dan konstanta untuk masing-masing *framework*. Nilai-nilai ini membentuk model persamaan regresi yang digunakan untuk memprediksi tren jumlah publikasi.

### 3.6 Perancangan Antarmuka

Antarmuka sistem berperan sebagai jembatan komunikasi antara pengguna dan sistem. Melalui antarmuka ini, pengguna dapat memasukkan dataset, mengelola data, hingga melihat hasil analisis tren dan prediksi penggunaan *framework*. Desain antarmuka disusun untuk memastikan bahwa proses input data hingga penyajian hasil analisis dapat dilakukan dengan lebih efektif dan terstruktur. Berikut ditampilkan rancangan desain antarmuka yang digunakan dalam sistem ini.

#### a. Halaman *Login*

Halaman *login* berfungsi sebagai pintu masuk bagi pengguna untuk mengakses sistem. Pengguna diwajibkan mengisi *username* dan *password* pada *form* yang tersedia, kemudian menekan tombol *login* untuk masuk ke dalam sistem dan mulai menggunakan fitur-fitur yang ada.



The image shows a login form with a rounded rectangular border. At the top center, the word "LOGIN" is displayed. Below it, there are two input fields: the first is labeled "Username" and the second is labeled "Password". At the bottom center of the form, there is a button labeled "Login".

**Gambar 3. 14** Tampilan Halaman *Login*

b. Halaman *Dashboard*

Halaman *dashboard* menjadi tampilan utama setelah pengguna berhasil *login*. Pada halaman ini, pengguna disambut dengan pesan selamat datang dan dapat melihat visualisasi hasil analisis berupa grafik yang menggambarkan tren penggunaan *framework*.

PREDIKSI TREN PENGUNAAN FRAMEWORK	
	<b>DASHBOARD</b>
Dashboard	Halo, User. Selamat Datang di Website Prediksi Tren Penggunaan Framework
Dataset	Hasil Analisis
Perhitungan	Grafik Hasil Analisis
Logout	Grafik Hasil Analisis

**Gambar 3. 15** Tampilan Halaman *Dashboard*

c. Halaman *Dataset*

Halaman *dataset* berfungsi untuk mengelola data jumlah publikasi *framework*. Di halaman ini, pengguna dapat menambah, mengedit, maupun menghapus data publikasi per tahun untuk masing-masing *framework*, serta dapat menambahkan *framework* baru jika diperlukan.

PREDIKSI TREN PENGUNAAN FRAMEWORK													
	<b>DATASET</b>												
	V Nama Framework												
Dashboard	Tambah Data												
Dataset													
Perhitungan	<table border="1"> <thead> <tr> <th>Tahun</th> <th>Jumlah Publikasi</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>20xx</td> <td>1234</td> <td><a href="#">Edit</a> <a href="#">Hapus</a></td> </tr> <tr> <td>20xx</td> <td>4321</td> <td><a href="#">Edit</a> <a href="#">Hapus</a></td> </tr> <tr> <td>20xx</td> <td>2314</td> <td><a href="#">Edit</a> <a href="#">Hapus</a></td> </tr> </tbody> </table>	Tahun	Jumlah Publikasi	Aksi	20xx	1234	<a href="#">Edit</a> <a href="#">Hapus</a>	20xx	4321	<a href="#">Edit</a> <a href="#">Hapus</a>	20xx	2314	<a href="#">Edit</a> <a href="#">Hapus</a>
Tahun	Jumlah Publikasi	Aksi											
20xx	1234	<a href="#">Edit</a> <a href="#">Hapus</a>											
20xx	4321	<a href="#">Edit</a> <a href="#">Hapus</a>											
20xx	2314	<a href="#">Edit</a> <a href="#">Hapus</a>											
Logout													
	+ Tambah Framework												

**Gambar 3. 16** Tampilan Halaman *Dataset*

#### d. Halaman Perhitungan

Halaman perhitungan menampilkan daftar *framework* yang tersedia dalam sistem. Pada halaman ini, pengguna dapat memilih *framework* tertentu, lalu menekan “Proses Perhitungan” yang akan mengarahkan pengguna ke halaman proses perhitungan untuk melihat detail proses analisis prediksi tren.

PREDIKSI TREN PENGUNAAN FRAMEWORK			
	<b>PERHITUNGAN</b>		
Dashboard	No	Framework	Proses
Dataset	1	Nama Framework	<a href="#">Proses Perhitungan</a>
Perhitungan	2	Nama Framework	<a href="#">Proses Perhitungan</a>
Logout	3	Nama Framework	<a href="#">Proses Perhitungan</a>

**Gambar 3. 17** Tampilan Halaman Perhitungan

#### e. Halaman Proses Perhitungan

Halaman proses perhitungan menampilkan proses analisis regresi linier secara rinci, mulai dari tabel data perhitungan hingga hasil prediksi dan nilai *error* yang diperoleh. Halaman ini membantu pengguna memahami proses dan hasil prediksi secara transparan.

PREDIKSI TREN PENGUNAAN FRAMEWORK						
	<b>PROSES PERHITUNGAN</b>					
Dashboard	Tahun	Jumlah Publikasi	X	Y	XY	X <sup>2</sup>
Dataset	20xx	1234	1	1234	X*Y	X*X
Perhitungan	20xx	4321	2	4321	X*Y	X*X
Logout	20xx	2314	3	2314	X*Y	X*X
	Total		$\Sigma X$	$\Sigma Y$	$\Sigma XY$	$\Sigma X^2$
	Koefisien (b)			Nilai b		
	Konstanta (a)			Nilai a		
	Hasil Prediksi					
	Tahun	Data Aktual	Data Prediksi			
	20xx	1234	Prediksi 1			
	20xx	4321	Prediksi 2			
	20xx	2314	Prediksi 3			
	MAE		Nilai MAE			

**Gambar 3. 18** Tampilan Halaman Proses Perhitungan

### 3.7 Perencanaan Pengujian Sistem

Pengujian terhadap hasil prediksi dilakukan dengan menggunakan model MAE untuk menentukan hasil akurasi dari data dengan menggunakan model persamaan regresi masing-masing *framework*. Perhitungan MAE dilakukan dengan cara mencari rata-rata dari selisih absolut antara nilai aktual dan nilai hasil prediksi pada periode tahun 2021 hingga 2024 yang berperan sebagai data uji. Perhitungan dilakukan menggunakan persamaan (2.2).

Pengujian sistem dilakukan menggunakan Pengujian *black box*. Metode ini merupakan metode yang berfokus pada pengujian fungsionalitas sistem tanpa memperhatikan struktur internal program. Pengujian dilakukan dengan memasukkan input tertentu dan mengevaluasi output yang dihasilkan, apakah sudah sesuai dengan kebutuhan.

**Tabel 3. 15** Pengujian *Black Box*

No	Pengujian	Cara Pengujian
1	Login Pengguna	<ol style="list-style-type: none"> <li>1. Pengguna mengisi <i>username</i> dan <i>password</i> di halaman Login.</li> <li>2. Pengguna menekan tombol “Login”.</li> <li>3. Sistem melakukan validasi <i>login</i>.</li> <li>4. Jika data valid, sistem mengarahkan pengguna ke halaman dashboard. Jika tidak, sistem menampilkan pesan kesalahan.</li> </ol>
2	Registrasi Pengguna	<ol style="list-style-type: none"> <li>1. Pengguna mengisi data yang dibutuhkan untuk membuat akun di halaman <i>Register</i>.</li> <li>2. Pengguna menekan tombol “Register”.</li> <li>3. Sistem menyimpan data dan mengarahkan pengguna ke halaman login.</li> </ol>

3	Dataset Publikasi	<ol style="list-style-type: none"> <li>1. Pengguna masuk ke menu “Dataset”.</li> <li>2. Pengguna dapat menekan tombol “Tambah Framework” untuk menambahkan <i>framework</i> baru.</li> <li>3. Pengguna dapat mengedit atau menghapus <i>framework</i> yang sudah ditambahkan.</li> <li>4. Setelah <i>framework</i> tersedia, pengguna dapat menekan tombol “Tambah Data” untuk menambahkan dataset publikasi.</li> <li>5. Pengguna dapat mengubah data publikasi dengan tombol “Edit”.</li> <li>6. Pengguna dapat menghapus data publikasi dengan tombol “Hapus”.</li> </ol>
4	Pembagian Dataset	<ol style="list-style-type: none"> <li>1. Pengguna memilih <i>framework</i> yang telah ditambahkan.</li> <li>2. Sistem menampilkan opsi pembagian dataset menjadi data latih dan data uji.</li> <li>3. Pengguna memilih jumlah tahun untuk data latih.</li> <li>4. Sistem memproses pembagian data dan menampilkan proses perhitungan.</li> </ol>
5	Proses Perhitungan Model	<ol style="list-style-type: none"> <li>1. Sistem menghitung koefisien (b) dan konstanta (a) untuk membentuk model regresi linier sederhana.</li> <li>2. Sistem menyimpan model regresi dan menampilkan model persamaan.</li> </ol>
6	Menampilkan Hasil Prediksi	<ol style="list-style-type: none"> <li>1. Sistem menggunakan model regresi yang sudah disimpan untuk menghitung nilai prediksi berdasarkan data uji.</li> </ol>

		2. Hasil prediksi ditampilkan dalam bentuk tabel perbandingan dengan data aktual.
7	Menampilkan Nilai <i>Error</i> (MAE)	<ol style="list-style-type: none"> <li>1. Sistem menghitung nilai <i>error</i> antara prediksi dan data aktual menggunakan rumus MAE.</li> <li>2. Sistem menampilkan nilai MAE dalam bentuk angka rata-rata selisih absolut di bawah tabel hasil prediksi.</li> </ol>

### 3.8 Spesifikasi Pengembangan Sistem

Spesifikasi pengembangan sistem berfungsi untuk menggambarkan kebutuhan teknis yang digunakan selama proses perancangan dan implementasi sistem. Spesifikasi ini mencakup perangkat keras (*hardware*) dan perangkat lunak (*software*) yang mendukung pengembangan, pengujian, serta pengoperasian sistem agar dapat berjalan secara optimal. Berikut ini adalah rincian spesifikasi yang digunakan dalam pengembangan sistem :

#### A. Perangkat Keras

Perangkat keras merupakan komponen fisik yang digunakan untuk mendukung proses pengembangan sistem. Pada sistem ini, perangkat keras yang dimanfaatkan adalah sebuah laptop dengan spesifikasi :

- a. Processor AMD A9
- b. RAM 4 GB
- c. HDD 1 TB
- d. Monitor 14"

#### B. Perangkat Lunak

Perangkat lunak adalah aplikasi yang digunakan dalam proses pengembangan sistem. Adapun perangkat lunak yang diperlukan untuk membangun sistem ini adalah:

- a. Windows 10

- b. Bahasa Pemrograman HTML, CSS, dan PHP
- c. Visual Studio Code
- d. *Web* Server Apache
- e. *Database* Server MySQL
- f. Composer

