

**MENENTUKAN TINGKAT KERUSAKAN SEKTOR PASCA  
BENCANA MENGGUNAKAN METODE TOPSIS DENGAN  
*QUERY OPTIMIZATION***

**SKRIPSI**

**OLEH :  
ANNISA HEPARYANTI SAFITRI  
NIM. 17650042**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**MENENTUKAN TINGKAT KERUSAKAN SEKTOR PASCA BENCANA  
MENGUNAKAN METODE TOPSIS DENGAN QUERY  
OPTIMIZATION**

**SKRIPSI**

**Diajukan kepada:  
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan  
Dalam Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:  
ANNISA HEPARYANTI SAFITRI  
NIM. 17650042**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG**

**2021**

**HALAMAN PERSETUJUAN**

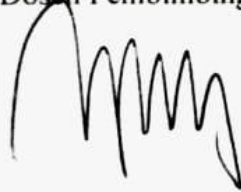
**MENENTUKAN TINGKAT KERUSAKAN SEKTOR PASCA BENCANA  
MENGUNAKAN METODE TOPSIS DENGAN *QUERY OPTIMIZATION***

**SKRIPSI**

Oleh:  
**ANNISA HEPARYANTI SAFITRI**  
NIM. 17650042

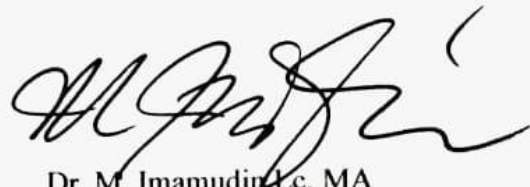
Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal: 26 November 2021

Dosen Pembimbing I



Agung Teguh Wibowo Almais, M.T  
NIP. 19860103 20180201 1 235


Dosen Pembimbing II



Dr. M. Imamudin, MA  
NIP. 19740602 200901 1 010

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Maulana Malik Ibrahim Malang



  
Dr. Fachrud Kurniawan ST., M.MT., IPM  
NIP. 19771020 200912 1 001

## HALAMAN PENGESAHAN

### MENENTUKAN TINGKAT KERUSAKAN SEKTOR PASCA BENCANA MENGGUNAKAN METODE TOPSIS DENGAN *QUERY OPTIMIZATION*

Oleh:  
**ANNISA HEPARYANTI SAFITRI**  
**NIM. 17650042**

Telah Dipertahankan di Depan Dewan Penguji  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)  
Pada Tanggal: 26 November 2021

#### Susunan Dewan Penguji:

Penguji Utama : Yunifa Miftachul Arif, M. T  
NIP. 19830616 201101 1 004

Ketua Penguji : Dr. M. Amin Hariyadi  
NIP. 19670018 200501 1 001

Sekretaris Penguji : Agung Teguh Wibowo Almais, M.T  
NIP. 19860103 20180201 1 235

Anggota Penguji : Dr. M. Imamudin Lc, MA  
NIP. 19740602 200901 1 010

#### Tanda Tangan

()

()

()


Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Negeri Islam Maulana Malik Ibrahim Malang



  
Dr. Fachrud Kurniawan ST., M.MT., IPM

NIP. 19771020 200912 1 001

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Annisa Heparyanti Safitri  
NIM : 17650042  
Fakultas : Sains dan Teknologi  
Jurusan : Teknik Informatika  
Judul Skripsi : Menentukan Tingkat Kerusakan Sektor Pasca  
Bencana Menggunakan Metode TOPSIS dengan  
*Query Optimization*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 10 November 2021

Yang membuat pernyataan.



Annisa Heparyanti Safitri  
NIM. 17650042

## **HALAMAN MOTO**

*“Work Hard, Dream Big”*

## HALAMAN PERSEMBAHAN

### الحمد لله رب العالمين

Tidak ada kata yang bisa diucapkan selain rasa beryukur penulis kepada Allah SWT yang telah memberikan nikmat Kesehatan sehingga dalam pengerjaan skripsi ini penulis mendapatkan kelancaran sampai terselesaikannya sampai tahap sidang akhir skripsi ini.

Tak lupa ucapan Terimakasih kepada keluarga penulis khususnya kedua orang tua yakni Bapak Adiansyah, S.Pd dan Ibu Siti Rahmawati S.Pd yang selalu memberikan doa, memberikan nasihat dan support sehingga penulis dapat sampai kepada tahap ini.

Serta rasa Terimakasih penulis terhadap Bapak Agung Teguh Wibowo Almais, M.T selaku dosen pembimbing dan Bapak Dr. M. Imamudin Lc, MA yang telah memberikan bimbingan berupa saran, motivasi dan dukungan sehingga saya bisa sampai ke tahap ini.

Dalam pembuatan skripsi ini tentunya ada teman di samping saya yang senantiasa membantu ketika saya berada dalam kesulitan, saya ucapkan terima kasih kepada Fenti Yulia Kristanti telah menjadi teman berfikir dan teman bertukar pikiran dikala saya mengalami hambatan ketika mengerjakan skripsi ini. Selain itu saya ucapkan terima kasih juga kepada teman-teman saya Rafika Syahrani, Nadila Sugianti dan Ayu Dian karena sering menghibur saya disaat saya sedang kesulitan selama mengerjakan skripsi. Semoga selalu dilancarkan dalam segala urusannya. آمين

## KATA PENGANTAR

Dalam perjalanan penulis menyelesaikan skripsi yang berjudul “Implementasi *Intelligence Decision Support System Dynamic* Untuk Menentukan Tingkat Kerusakan Sektor Pasca Bencana Alam Menggunakan *Modified-TOPSIS* dan *Neural Network* “ ini tak henti-hentinya mengucap rasa syukur karena diberikan kelancaran dan kemudahan.

Skripsi ini telah berhasil diselesaikan oleh penulis atas bantuan dari beberapa pihak. Sehingga penulis ingin menyampaikan ucapan terimakasih kepada:

1. Bapak Agung Teguh Wibowo Almais, M.T yang telah memberikan ilmu yang bermanfaat bagi saya dan selalu mendukung di setiap langkah pengerjaan skripsi ini.
2. Dr. M. Imamudin Lc, MA yang telah memberikan ilmu yang bermanfaat kepada saya.
3. Bapak Yunifa Miftachul Arif, M. T dan Bapak Dr. M. Amin Hariyadi selaku dosen penguji yang telah memberikan bimbingan serta saran pada skripsi ini.
4. Ayah dan Ibu yang selalu memberikan motivasi dan doa serta mendukung setiap langkah saya.
5. Seluruh teman-teman Unocore yang telah memberikan semangat, dukungan serta informasi yang bermanfaat kepada penulis ketika mengalami kesulitan.
6. Terakhir, terimakasih untuk diri saya sendiri karena telah berhasil melewati semua kesulitan yang dihadapi sebelumnya dan tetap berusaha untuk hasil yang terbaik.

Penulis ucapkan terima kasih sebesar-besarnya kepada orang-orang tersebut. Semoga Allah SWT membalas seluruh perbuatan baik yang telah



mereka perbuat kepada penulis dan menjadi amal jariyah bagi mereka. Semoga karya ini dapat bermanfaat bagi teman-teman semua khususnya adik tingkat yang akan melakukan penelitian serupa. Semoga ilmu yang tertuang dalam tulisan ini dapat bermanfaat bagi generasi penerus bangsa.

Malang, 21 November 2021

Penulis

## DAFTAR ISI

HALAMAN PERSETUJUAN .....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN KEASLIAN TULISAN .....	iv
HALAMAN MOTO .....	v
HALAMAN PERSEMBAHAN .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiii
ABSTRAK .....	xiv
ABSTRACT.....	xv
المخلص .....	xvi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Pernyataan Masalah .....	6
1.3 Tujuan .....	6
1.4 Batasan Masalah.....	7
1.5 Manfaat Penelitian .....	7
BAB II STUDI LITERATUR.....	8
2.1 Penelitian Terkait .....	8
2.2 TOPSIS .....	14
2.3 <i>Query</i> optimization.....	16
2.4 Response Time .....	19
2.5 Usability .....	20
BAB III METODOLOGI PENELITIAN.....	22
3.1 Desain Sistem.....	22
3.2 Data Collection.....	24
3.3 Identifikasi Perangkat Keras dan Perangkat Lunak yang Digunakan .....	24

3.4 Alternatif dan Kriteria .....	25
3.5 Perhitungan Manual Metode TOPSIS.....	26
3.5.1 Membuat Matrik Keputusan .....	26
3.5.2 Matriks Ternormalisasi(R).....	26
3.5.3 Matriks Normalisasi Terbobot(Y) .....	27
3.5.4 Matriks Solusi Ideal Positif (A+).....	27
3.5.5 Matriks Solusi Ideal Positif (A-).....	27
3.5.6 Matriks Jarak Solusi Ideal Positif (D+) Jarak Solusi Ideal Negatif (D-) .....	28
3.5.7 Perangkingan .....	28
3.6 Perbandingan <i>Query</i> optimization dan Non-Optimization .....	29
3.7 Rencana Pengujian .....	31
3.7.1 Membandingkan Dua Jenis <i>Query</i> ( <i>Query</i> Non-Optimization dan <i>Query</i> Optimization).....	31
3.7.2 Pengujian Menggunakan Response Time.....	32
3.8 Implementasi Sistem .....	33
BAB IV HASIL DAN PEMBAHASAN .....	38
4.1 Implementasi <i>Query Optimization</i> .....	38
4.2 Uji Coba <i>Query Optimization</i> .....	45
4.2.1 Uji Coba Ke-1 .....	47
4.2.2 Uji Coba ke-2.....	49
4.2.3 Uji Coba Ke-3.....	51
4.2.4 Perbandingan Hasil Uji Coba .....	54
4.2.5 Hasil Uji Coba Usability.....	55
4.3 Integrasi Islam.....	57
BAB V KESIMPULAN DAN SARAN.....	60
5.1 Kesimpulan .....	60
5.2 Saran.....	60
DAFTAR PUSTAKA .....	622

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Flowchart Metode TOPSIS .....	16
<b>Gambar 2. 2</b> Flowchart Metode TOPSIS ( <i>Query Optimization</i> ).....	18
<b>Gambar 3. 1</b> Blok Diagram Sistem.....	23
<b>Gambar 3. 2</b> Halaman Login .....	33
<b>Gambar 3. 3</b> Halaman Super Admin Data Kriteria .....	34
<b>Gambar 3. 4</b> Halaman Super Admin Data Alternatif .....	34
<b>Gambar 3. 5</b> Halaman Super Admin Data Sektor .....	35
<b>Gambar 3. 6</b> Halaman Super Admin Jenis Bencana.....	35
<b>Gambar 3. 7</b> Halaman Admin Weight .....	36
<b>Gambar 3. 8</b> Halaman Admin TOPSIS .....	36
<b>Gambar 3. 9</b> Halaman Admin Topsis QO dan Topsis Non-QO.....	37
<b>Gambar 3. 10</b> Halaman Admin Hasil Alternatif.....	37
<b>Gambar 4. 1</b> Desain Database.....	39
<b>Gambar 4. 2</b> Response Time Tabel 2normalisasi .....	46
<b>Gambar 4. 3</b> Response Time Tabel 2normalisasi_optimasi .....	46
<b>Gambar 4. 4</b> Grafik Uji Coba ke-1 .....	48
<b>Gambar 4. 5</b> Grafik Uji Coba Ke-2 .....	50
<b>Gambar 4. 6</b> Grafik Uji Coba Ke-3 .....	52

## DAFTAR TABEL

<b>Tabel 2. 1</b> Daftar Pertanyaan SUS.....	20
<b>Tabel 3. 1</b> Spesifikasi Perangkat Lunak yang Digunakan.....	24
<b>Tabel 3. 2</b> Spesifikasi Perangkat Keras yang Digunakan.....	24
<b>Tabel 3. 3</b> Alternatif .....	25
<b>Tabel 3. 4</b> Kriteria.....	25
<b>Tabel 3. 5</b> Perhitungan Matriks Keputusan .....	26
<b>Tabel 3. 6</b> Matriks Ternormalisasi.....	27
<b>Tabel 3. 7</b> Matriks Normalisasi Terbobot .....	27
<b>Tabel 3. 8</b> Matriks Solusi Ideal Positif (A+) .....	27
<b>Tabel 3. 9</b> Matriks Solusi Ideal Positif (A-) .....	28
<b>Tabel 3. 10</b> Matriks D+ dan D.....	28
<b>Tabel 3. 11</b> Perangkingan Beserta Alternatif .....	28
<b>Tabel 4. 1</b> Perbandingan Query Optimization dan Non-Optimization.....	40
<b>Tabel 4. 2</b> Jumlah data untuk uji coba.....	45
<b>Tabel 4. 3</b> Tabel uji coba ke-1 .....	47
<b>Tabel 4. 4</b> Tabel Uji coba ke-2 .....	49
<b>Tabel 4. 5</b> Tabel Uji Coba Ke-3 .....	51
<b>Tabel 4. 6</b> Perbandingan Hasil Uji Coba.....	54
<b>Tabel 4. 7</b> Hasil Penilaian Skor SUS.....	55

## ABSTRAK

Annisa Heparyanti Safitri. 2021. *Menentukan Tingkat Kerusakan Sektor Pasca Bencana Menggunakan Metode TOPSIS dengan Query Optimization*. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing : (I) Agung Teguh Wibowo Almais, M.T (II) Dr. M. Imamudin Lc, MA

---

Kata Kunci : *Query Optimization*, TOPSIS, response time, Usability

Volume data yang sangat besar dari tim surveyor Perencanaan dan Pengendalian Penanganan Bencana(P3B) menciptakan masalah yang luas dan beragam sehingga dapat menghabiskan sumber daya sistem dan waktu pemrosesan yang terbilang lama. Oleh karena itu penelitian ini mengusulkan solusi dengan melakukan optimasi *query* pada metode setiap tahap metode TOPSIS yang diimplementasikan pada sistem pendukung keputusan untuk menentukan tingkat kerusakan pasca bencana.. Berdasarkan 3 kali uji coba dengan jumlah data yang berbeda-beda yaitu ujicoba ke-1 menggunakan 114 data, ujicoba ke-2 sebanyak 228 data dan ujicoba ke-3 menggunakan 334 data didapati hasil uji coba yang menunjukkan pengoptimalan pada *query* berhasil dilakukan. Hal ini ditandai dengan nilai *response time* pada *query* yang dioptimasi lebih kecil dibandingkan dengan *query* yang tidak dioptimasi. Selain itu, hasil dari *usability* tes menggunakan SUS menunjukkan skor 50.16667 menunjukkan bahwa sistem masih belum *usable*. Hal ini menunjukkan bahwa sistem perlu dikembangkan lebih lanjut.

## ABSTRACT

Annisa Heparyanti Safitri. 2021. *Determining the Damage Level of the Post-Disaster Sector Using the TOPSIS Method with Query Optimization*. Theses. Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University Malang. Advisor : (I) Agung Teguh Wibowo Almais, M.T (II) Dr. M. Imamudin Lc, MA

---

Keywords: *Query* Optimization, TOPSIS, response time, Usability

The huge volume of data from the Disaster Management Planning and Control (P3B) surveyor team creates wide and varied problems that can consume system resources and process time that is relatively long. Therefore, this study proposes a solution by performing *query* optimization on the method of each stage of the TOPSIS method which is implemented in a decision support system to determine the level of post-disaster damage. Based on 3 trials with different amounts of data, namely the 1st trial using 114 data, the 2nd trial of 228 data and the 3rd trial using 334 data, the test results showed that the optimization of the *query* was successfully carried out. This is indicated by the response time value in the optimized *query* is smaller than the non-optimized *query*. In addition, the results of the usability test using SUS showed a score of 50,16667 indicating that the system was still not usable. This shows that the system needs to be developed further.

## الملخص

النساء هيباريانتي سافيتري. 2021. تحديد مدى الضرر لقطاع ما بعد الكارثة باستخدام طريقة *TOPSIS* مع تحسين الاستعلام. البحث الجامعي. قسم هندسة المعلوماتية بكلية العلوم والتكنولوجيا جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: أجونج تيجوه ويوو الميس الماجستير، المشرف الثاني: الدكتور محمد إمام الدين الماجستير.

---

الكلمات الرئيسية: تحسين الاستعلام ، *TOPSIS* ، وقت الاستجابة ، سهولة الاستخدام

حجم البيانات الضخم من فريق مساح الأراضي لتخطيط إدارة الكوارث والتحكم ينتج المشاكل الواسعة والمتنوعة حتى تستهلك موارد النظام ووقت العملية الطويل. لذلك ، يقترح هذا البحث حلاً بطريق أداء تحسين الاستعلام في كل مرحلة من مراحل طريقة *TOPSIS* التي تطبقها في نظام دعم القرار لتحديد مدى الضرر بعد وقوع الكارثة. استناداً إلى 3 مرات للتجربة بعدد البيانات المختلفة ، أي التجربة الأولى باستخدام 114 بيانات ، والتجربة الثانية باستخدام 228 بيانات والتجربة الثالثة باستخدام 334 بيانات ، أظهرت نتائج الاختبار على أن تحسين الاستعلام قد حدث بنجاح. يشار إلى ذلك من خلال قيمة وقت الاستجابة في الاستعلام المحسنة أصغر من الاستعلام غير المحسن. إضافة إلى ذلك ، تظهر نتائج اختبار سهولة الاستخدام باستخدام *SUS* على درجة 50،16667 تشير إلى أن النظام لا يزال غير قابل للاستخدام. هذا يدل على أن النظام يحتاج إلى مزيد من التطوير.



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Bencana alam yang sering terjadi di Indonesia diantaranya yaitu banjir, kebakaran, puting beliung, dan tanah longsor. Sebagian besar setiap provinsi di Indonesia mempunyai catatan bencana alam yang pernah terjadi. Dampak dari kejadian bencana alam tersebut sering sekali menyebabkan kerugian berupa harta benda hingga korban jiwa ( Yana, 2018 ). Kurang lebih 13% dari gunung berapi dunia yang bertempat di Indonesia memiliki potensi untuk memicu bencana dengan kekuatan dan intensitas yang berbeda. Penanggulangan bencana alam merupakan upaya selanjutnya untuk mengurangi dampak dari bencana alam terhadap manusia dan harta benda ( Sadewo, 2018).

Instansi pemerintah yang memiliki tanggung jawab untuk menangani terkait penanggulangan bencana yaitu Badan Nasional Penanggulangan Bencana atau dikenal juga dengan BNPB yang merupakan badan yang bersifat non struktural untuk penanggulangan bencana yang berada di bawah Presiden dan bertanggung jawab langsung kepada Presiden. Penanggulangan yang ditangani oleh BPBD berupa pra maupun pasca bencana alam (Mahdia, 2013). Persiapan dalam menangani bencana alam sendiri meliputi aktivitas yang dapat diusahakan sebelum terdeteksinya tanda-tanda bencana sehingga dapat membantu dalam pemakaian sumber daya yang tersedia, meminta bala bantuan serta merencanakan rehabilitasi dengan cara dan kemungkinan yang paling baik (Sadewo, 2018).

Kesuksesan yang diperoleh tim Perencanaan dan Pengendalian Penanganan Bencana (P3B) dalam usaha pemulihan pascabencana sangat bergantung pada perencanaan pemulihan yang baik yang bersumber dari data dan informasi yang akurat. Jumlah permasalahan yang besar di lapangan diakibatkan oleh kurang akuratnya data. Keadaan ini terjadi karena kriteria pengumpulan data awal yang digunakan oleh surveyor di lapangan dipandang tidak sama sehingga kategorisasi data yang ada berbeda. Dengan demikian, timbul masalah dan data bencana yang masuk ke BPBD Provinsi berbeda dengan keadaan di lapangan karena kriteria data untuk persiapan tindakan rehabilitasi dan rekonstruksi pascabencana yang digunakan oleh masing-masing surveyor dianggap berbeda (Bachriwindi, 2019).

Dari identifikasi permasalahan yang sudah dijabarkan, diperlukan suatu aplikasi untuk membantu tim surveyor dalam proses persiapan tindakan rekonstruksi dan rehabilitasi pasca bencana. Penggunaan aplikasi yang dilengkapi dengan *Decision Support System*(DSS) diharapkan dapat mempersingkat waktu tim surveyor dalam melakukan tindakan rekonstruksi dan rehabilitasi pasca bencana alam, selain itu, bisa juga meningkatkan akurasi dalam mendapatkan hasil yang tepat (Almais, 2016). Namun, volume data yang sangat besar ini menciptakan masalah yang tidak terselesaikan untuk eksekusi *query* DSS. *Query* DSS yang luas dan beragam dapat menghabiskan sumber daya sistem dan waktu pemrosesan yang terbilang lama. Untuk mengoptimalkan sumber daya dan untuk mempercepat

pengambilan data klinis dan proses analisis, *query* DSS harus dioptimalkan secara efektif (Sharma, 2018).

Perkiraan pemrosesan *query* baru-baru ini muncul sebagai solusi yang layak untuk menangani sejumlah besar data, kompleksitas *query* yang tinggi, dan persyaratan *response time* yang semakin ketat yang menjadi ciri aplikasi sistem pendukung keputusan (DSS) saat ini. Biasanya, pengguna DSS memiliki sikap yang sangat kompleks kueri ke sistem manajemen basis data (DBMS) yang mendasari yang memerlukan operasi kompleks melalui Gigabyte atau Terabyte dari data penduduk disk dan, dengan demikian, membutuhkan waktu yang sangat lama untuk dieksekusi hingga selesai dan menghasilkan jawaban yang tepat (Chakrabarti, 2001).

Oleh karena itu penelitian ini mengusulkan solusi dengan melakukan Optiasi *query* terhadap *query* DSS. Optimasi *query* adalah tugas yang merangsang dari setiap sistem database. Sejumlah heuristik telah diterapkan belakangan ini, yang mengusulkan algoritma baru untuk secara substansial meningkatkan kinerja *query*. Pencarian solusi yang lebih baik masih terus berlanjut. Perkembangan yang tidak dapat hilang dalam bidang database Sistem Pendukung Keputusan (DSS) menyajikan data dengan kecepatan yang luar biasa (Sharma, 2016). Pengoptimalan *query* akhirnya menjadi tantangan terbesar bagi para peneliti database. Optimasi *query* dalam sistem database terdistribusi telah mendapatkan perhatian yang cukup besar dalam beberapa tahun terakhir. Ini adalah proses untuk menentukan rencana eksekusi *query* terbaik dalam hal *Response time*.

*Response time* yaitu waktu tanggap yang diberikan oleh interface ketika user mengirim request ke komputer. Secara umum, user menginginkan bahwa program aplikasinya bisa memberikan waktu tanggap yang sesingkat-singkatnya. Akan tetapi waktu tanggap yang ideal tidak dapat ditentukan, karena ada beberapa aspek yang mempengaruhi, seperti ragam interaksi yang diinginkan dan kelancaran pengguna dalam menjalankan program aplikasi tersebut (Noviyanti, 2018).

Pada penelitiannya, Purwanto (2017) menyebutkan bahwa Topsis mempunyai keunggulan dimana cara kerjanya yang tidak rumit dan mudah dipahami, komputasinya efisien dan mempunyai kelebihan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana. Disamping itu metode TOPSIS dapat mengatasi perbedaan alternatif meskipun perbedaannya terbilang kecil, dalam metode TOPSIS ada yang dinamakan dengan kaidah Cost dan Benefit yang digunakan untuk menentukan kaidah dalam setiap kriteria, oleh karena itu dengan kelebihan yang ada pada metode TOPSIS tersebut, maka penelitian ini akan menggunakan metode TOPSIS sebagai metode untuk penyelesaian masalah.

Untuk mengoptimalkan kinerja metode TOPSIS, maka dilakukan optimasi pada *query*nya sehingga diharapkan optimasi terhadap *query* metode TOPSIS pada DSS dapat mempersingkat waktu dalam menyelesaikan permasalahan yang telah dipaparkan sebelumnya. Dari optimasi ini akan didapatkan kelebihan untuk mempercepat proses pertolongan terhadap korban bencana dalam menata kembali kehidupan korban bencana setelah terdampak bencana

alam. Seperti pada salah satu potongan ayat Al-Quran surat Al-Maidah ayat 2, Allah SWT berfirman:

وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۚ وَاتَّقُوا اللَّهَ ۖ

إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

*“Dan tolong menolong lah kamu dalam kebaikan dalam kebaikan dan ketakwaan. Dan janganlah tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya siksa Allah sangat berat”.* (QS. Al-Maidah : 2)

Sesuai dengan tafsir Jalalain (Harry, 2003), QS. Al-Maidah ayat dua mengungkapkan bahwa (Bertolong-tolonglah engkau pada kebaikan) dalam mengerjakan yang ditahankan (dan ketakwaan) dengan meninggalkan apa-apa yang tidak boleh (serta janganlah kamu bertolong-tolongan) pada *ta`aawanu* dibuang keliru satu di antara dua ta pada Asalnya (dalam berbuat dosa) atau maksiat (serta pelanggaran) ialah melampaui batas-batas ajaran Allah. (serta bertakwalah engkau pada Allah) takutlah kamu kepada azab siksa-Nya dengan menaati-Nya (sesungguhnya Allah amat berat siksa-Nya) bagi orang yang menentang-Nya.

Penelitian ini akan menerepakan metode TOPSIS sebagai metode DSS. untuk mengoptimalkan metode TOPSIS yang digunakan, maka dilakukan optimasi pada *query* DSS. Optimasi *query* atau dikenal juga dengan *query optimization* ini sering menjadi penyelesaian masalah untuk mengurangi beban database dimana penggunaan sumber daya sistem yang diperlukan untuk mengeksekusi *query* menjadi perhatian utama. Oleh karena itu, *query* DSS dioptimasi berdasarkan *response time* sehingga diharapkan dapat

mempersingkat waktu pembacaan serta waktu eksekusi *query* DSS. Oleh karena itu, pada penelitian ini akan dilakukan perbandingan terhadap DSS *query* optimization dengan non *query* optimization sehingga dapat diketahui efektifitas dari optimasi *query* DSS ini.

Melalui penelitian ini juga, akan didapati perbedaan *response time* pada *query* TOPSIS yang tradisional dan *query* TOPSIS yang sudah melalui tahapan optimasi. Diharapkan optimasi ini dapat mempersingkat *response time* yang nantinya dapat mempercepat proses penanganan pasca bencana alam.

Terakhir, akan dilakukan pula penelitian untuk mengetahui nilai *usability* pada sistem yang dibangun dalam penelitian sehingga dapat diketahui kualitas dan efektifitas sistem yang sudah dibangun.

## **1.2 Pernyataan Masalah**

Berdasarkan penjelasan latar belakang pada sub bab sebelumnya, maka dapat dinyatakan permasalahan dalam penelitian ini adalah bagaimana mengukur *response time query* optimization untuk diterapkan pada DSS untuk menentukan tingkat kerusakan sektor pasca bencana alam menggunakan metode TOPSIS dan berapa nilai usabilitasnya ?

## **1.3 Tujuan**

Mengetahui *response time query* optimization yang diterapkan pada DSS untuk menentukan tingkat kerusakan sektor pasca bencana alam menggunakan metode TOPSIS.

#### **1.4 Batasan Masalah**

Menggunakan data Pusat Vulkanologi dan Mitigasi Bencana Geologi (Badan Nasional Penanggulangan Bencana, 2019) di Provinsi Jawa Timur.

#### **1.5 Manfaat Penelitian**

Adapun manfaat penelitian ini antara lain :

1. Membantu surveyor lebih cepat menentukan tingkat kerusakan sektor pasca bencana alam.
2. Mengetahui respons time saat eksekusi *query* DSS untuk menentukan tingkat kerusakan sektor pasca bencana alam menggunakan metode TOPSIS.

## **BAB II**

### **STUDI LITERATUR**

#### **2.1 Penelitian Terkait**

Penelitian yang dilakukan oleh Almais (2016) menjelaskan bahwa penelitiannya bertujuan untuk mengusulkan suatu metode yang menggunakan asesmen Non Numerik yaitu *Multi Experts Multi Criteria Decision Making* (MEMCDM) dalam rangka membangun sistem keputusan pendukung dalam penyusunan tindakan rehabilitasi dan rekonstruksi pasca bencana. Dikarenakan, tim surveyor Perencanaan dan Pengendalian Penanganan Bencana (P3B) tidak mempunyai kriteria standar yang jelas untuk melaksanakan penyusunan aksi rehabilitasi dan rekonstruksi pasca bencana. Metode Pengambilan Keputusan Multi Pakar Multi Kriteria (MEMCDM) adalah menentukan alternatif, skala penilaian, kriteria, kualitas kriteria, negasi kualitas kriteria, agregasi kriteria, dan skor pakar kualifikasi. Pada tahapan skor kualifikasi ahli untuk data primer dan sekunder berbeda. Pada data sekunder, skor kualifikasi didasarkan pada penggunaan jumlah ahli.

Bachriwindi (2019) dalam penelitiannya mengungkapkan bahwa penentuan kerusakan dan kerugian pasca bencana alam dilakukan untuk menentukan berbagai macam kerusakan serta jumlah kerugian pasca bencana alam yang harus ditangani oleh pemerintah. Agar berbagai macam kerusakan dan jumlah kerugian setelah bencana alam samas dengan data di lapangan, maka dilakukan studi yang menerapkan *Decision Support System* (DSS) dengan menggunakan metode *Weighted Product* (WP). Hasil yang diperoleh berupa *testing* data pola yang



diukur dengan menggunakan *Weighted Product* (WP) dibandingkan dengan 3 data *testing* yang berbeda yaitu dari data kerusakan dan kerugian pasca bencana alam di Jawa Timur tahun 2010 tahun 2011.

Almais (2020) berpendapat bahwa penyusunan tindakan rehabilitasi rekonstruksi pasca bencana dilakukan agar mendapatkan tingkat kerugian dan kerusakan pasca bencana yang harus ditangani oleh pemerintah. Guna mencegah tingkat kerusakan dan kerugian pasca bencana yang sesuai dengan kondisi di lapangan maka dilakukan penelitian penerapan *Decision Support System Dynamic* (DSSD) dengan menggunakan metode *Fuzzy-Weighted Product* (F-WP). *Decision Support System Dynamic* (DSSD) merupakan pengembangan dari model *Decision Support System* (DSS) terbaru, sedangkan *Fuzzy* merupakan algoritma untuk menentukan tingkat kepentingan setiap kriteria yang digunakan dalam metode *Weighted Product*. Sedangkan *Weighted Product* digunakan sebagai pembentukan data pola sistem.

Penelitian selanjutnya yang dilakukan oleh Kuo (2017) mengungkapkan bahwa sebagai alat analisis keputusan, TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*) berupaya untuk memilih alternatif yang secara simultan memiliki jarak terdekat dari solusi ideal positif (PIS) dan jarak terjauh dari solusi ideal negatif (NIS). Meskipun indeks peringkat TOPSIS masuk akal, namun ada kekurangan. Artinya, indeks peringkat ini terlepas dari bobot pemisahan alternatif dari PIS dan NIS. Dengan kata lain, tidak peduli berapa bobot yang diberikan oleh pembuat keputusan untuk dua pemisahan ini, hasil

pemeringkatan tidak akan berbeda seolah-olah dia tidak memiliki preferensi untuk kedua pemisahan ini. Cacat ini tentunya akan membatasi penerapan TOPSIS.

Schwarz (2001) dalam penelitiannya membahas alat yang menghasilkan urutan pernyataan SQL untuk menghasilkan informasi yang diminta. Analisis menyeluruh mengungkapkan bahwa banyak urutan *query* yang dihasilkan oleh alat komersial tidak terlalu efisien. Arsitektur sistem yang dioptimalkan disarankan untuk aplikasi ini. Komponen utamanya adalah pengoptimal DSS yang menerima urutan *query* yang dibuat sebelumnya dan memodelkannya kembali sesuai dengan serangkaian strategi pengoptimalan, sebelum dijalankan oleh sistem database yang mendasarinya. Keuntungan dari arsitektur yang diperluas ini dibahas dan beberapa strategi pengoptimalan yang sesuai diidentifikasi.

Pada penelitian yang dilakukan oleh Sharma (2016) menjelaskan bahwa optimasi *query* adalah tugas yang merangsang dari setiap sistem database. Sejumlah heuristik telah diterapkan belakangan ini, yang mengusulkan algoritma baru untuk secara substansial meningkatkan kinerja *query*. Pencarian solusi yang lebih baik masih terus berlanjut. Perkembangan yang tidak dapat hilang dalam bidang database Sistem Pendukung Keputusan (DSS) menyajikan data dengan kecepatan yang luar biasa. Volume data DSS yang sangat besar hanya menjadi konsekuensi jika data tersebut dapat diakses dan dianalisis oleh peneliti yang berbeda. Di sini, kerangka kerja stokastik inovatif dari pengoptimal *query* DSS diusulkan untuk lebih mengoptimalkan desain pendekatan genetik pengoptimalan *query* yang ada.

Pada penelitian lainnya Sharma (2018) mengungkapkan bahwa terdapat dua teknik komputasi yang terinspirasi oleh alam, yaitu Firefly dan Algoritma Genetik terkontrol dalam lingkungan divergensi terbatas telah digabungkan untuk mengusulkan pengoptimalan *query* Sistem Pendukung Keputusan Klinis (CDSS) yang ditingkatkan. Model yang diusulkan dimaksudkan untuk mendapatkan rencana eksekusi *query* yang lebih baik yang akan mengurangi input-output, pemrosesan dan komunikasi untuk eksekusi *query* CDSS. Penggunaan gabungan kerangka pengoptimalan *query* CDSS yang diusulkan akan menghasilkan variasi substansial dalam dua generasi berturut-turut yang secara efektif akan menyelesaikan masalah konvergensi yang lambat dari Algoritma Genetik yang dikendalikan. Selanjutnya, hasil pengoptimalan *query* CDSS yang diusulkan diuji terhadap efek pengoptimalan *query* CDSS berbasis algoritma genetik lainnya.

Selanjutnya, ada penelitian yang dilakukan oleh Eslami (2020) yang menjelaskan bahwa dalam studinya ini, mereka mengembangkan Mixed Binary Quadratic Program (MBQP) untuk mengoptimalkan *query*-batching dalam sistem database. Model ini menggunakan koefisien regresi linier pada waktu pengambilan *query*, dilatih oleh sekumpulan besar kumpulan *query* yang dibuat secara acak. Peneliti juga mengusulkan dua heuristik, yang disebut metode pencarian kardinalitas terbatas I dan II, untuk menyelesaikan MBQP yang diusulkan. Untuk mendemonstrasikan keefektifan teknik yang peneliti usulkan, peneliti melakukan studi komputasi yang komprehensif atas contoh yang dibuat secara acak dari tiga benchmark database terkenal.

Berikutnya, berdasarkan penelitian yang dilakukan oleh Chakrabarti (2001) yang menjelaskan bahwa perkiraan pemrosesan *query* telah muncul sebagai pendekatan hemat biaya untuk menangani volume data yang besar dan persyaratan *response time* yang ketat dari sistem pendukung keputusan (DSS) saat ini. Namun, sebagian besar pekerjaan di area ini sejauh ini terbatas dalam lingkup pemrosesan *query*, biasanya berfokus pada bentuk *query* agregat tertentu. Lebih lanjut, pendekatan konvensional berdasarkan pengambilan sampel atau histogram tampaknya secara inheren terbatas ketika mendekati hasil *query* kompleks melalui kumpulan data DSS dimensi tinggi. Dalam makalah ini, peneliti mengusulkan penggunaan wavelet multi-dimensi sebagai alat yang efektif untuk pemrosesan perkiraan tujuan umum dalam aplikasi modern berdimensi tinggi.

Ada pula penelitian yang dilakukan oleh Liang (2011) menjelaskan bahwa dalam penelitiannya mereka mempelajari masalah *query* top-k dengan batasan *response time* dalam jaringan sensor nirkabel. Pertama-tama mereka membahas *response time query* dan pengaruhnya pada masa jaringan secara eksplisit. Kemudian mengusulkan kerangka kerja pengoptimalan *query* gabungan baru yang terdiri dari menemukan pohon perutean di jaringan dan merancang algoritme untuk evaluasi *query* teratas pada pohon tersebut dengan cara yang efisien untuk memenuhi batasan *response time* yang ditentukan. Pada akhirnya melakukan eksperimen ekstensif melalui simulasi untuk mengevaluasi kinerja kerangka pengoptimalan yang diusulkan, dalam hal konsumsi energi total, konsumsi energi maksimum di antara node, *response time query*, dan masa pakai jaringan. Hasil eksperimen menunjukkan bahwa ada tradeoff non-sepele antara *response time*

*query* dan masa pakai jaringan, dan kerangka kerja pengoptimalan bersama yang diusulkan dapat memperpanjang masa pakai jaringan secara signifikan di bawah *response time query* yang diberikan.

## 2.2 TOPSIS

TOPSIS telah diterapkan dalam banyak aspek termasuk perancangan robot, evaluasi pelanggan, perbandingan kinerja dari perusahaan, keputusan investasi keuangan, hingga pemilihan sistem operasi.

Teori ini mencakup baik alternatif ideal maupun alternatif ideal negatif, yang relatif komprehensif dan mudah dihitung. Selain itu, model ini sangat dapat ditafsirkan dan telah diterapkan di banyak arena praktik. Karenanya, TOPSIS menjadi hot spot yang menjadi perhatian para peneliti, terutama setelah milenium baru (Yu, 2021).

Berikut merupakan langkah - langkah dalam menyelesaikan sebuah kasus menggunakan metode TOPSIS (Febriyati, 2016):

1. Menyusun matriks keputusan R dengan menggunakan persamaan 2.1 berikut:

$$R = (r_{ij})_{m \times n} = \left( \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \right)_{m \times n} \quad ..(2.1)$$

2. Membuat matriks keputusan yang ternormalisasi dengan menggunakan persamaan 2.2 berikut:

$$D = (x_{ij})_{m \times n} \quad ..(2.2)$$

Dengan  $i=1,2,\dots,m$ ; nilai  $m$  menunjukkan jumlah alternatif yang dievaluasi, dan nilai  $X_{ij}$  menunjukkan nilai rating kecocokan alternatif ke- $i$  terhadap kriteria ke- $j$ .

3. Membangun matriks keputusan ternormalisasi berbobot V dengan menggunakan persamaan 2.3 berikut:

$$V = (v_{ij})_{m \times n} = (w_j r_{ij})_{m \times n} \quad ..(2.3)$$

Dengan  $w_j$  adalah pangkat bernilai positif untuk atribut keuntungan (Benefit), dan bernilai negatif untuk atribut biaya (cost). Nilai  $w_j$  menunjukkan nilai bobot dari kriteria C yang ke-j.

4. Identifikasi solusi ideal positif (PIS) ( $A^+ = \{v_1^+, v_2^+, \dots, v_n^+\}$ ) dan solusi ideal negatif (NIS) ( $A^- = \{v_1^-, v_2^-, \dots, v_n^-\}$ ) dengan menggunakan persamaan 2.4 dan 2.5 berikut:

$$v_j^+ = \{(\max_i v_{ij} | i \in J_b), (\min_i v_{ij} | j \in J_c) | i \in [1..m]\} \quad ..(2.4)$$

$$v_j^- = \{(\min_i v_{ij} | i \in J_b), (\max_i v_{ij} | j \in J_c) | i \in [1..m]\} \quad ..(2.5)$$

5. Hitung ukuran pemisahan ( $s_i^+$  dan  $s_i^-$ ) dengan menggunakan persamaan 2.6 dan 2.7 berikut:

$$s_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad \forall i \in [1..m] \quad ..(2.6)$$

$$s_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad \forall i \in [1..m] \quad ..(2.7)$$

6. Tentukan koefisien kedekatan ( $CC_i$ ) menggunakan persamaan 2.8 berikut:

$$CC_i = \frac{s_i^-}{(s_i^+ + s_i^-)} \quad \forall i \in [1..m] \quad ..(2.8)$$

Berikut langkah-langkah metode TOPSIS ditampilkan dalam bentuk

*flowchart* :



**Gambar 2. 1** *Flowchart* Metode TOPSIS

### 2.3 *Query optimization*

Dengan adanya *query*, ada banyak rencana akses yang dapat diikuti oleh sistem manajemen basis data (DBMS) untuk memprosesnya dan menghasilkan jawabannya. Semua rencana setara dalam hal hasil akhirnya tetapi biayanya bervariasi, yaitu jumlah waktu yang diperlukan untuk menjalankannya. Perbedaan

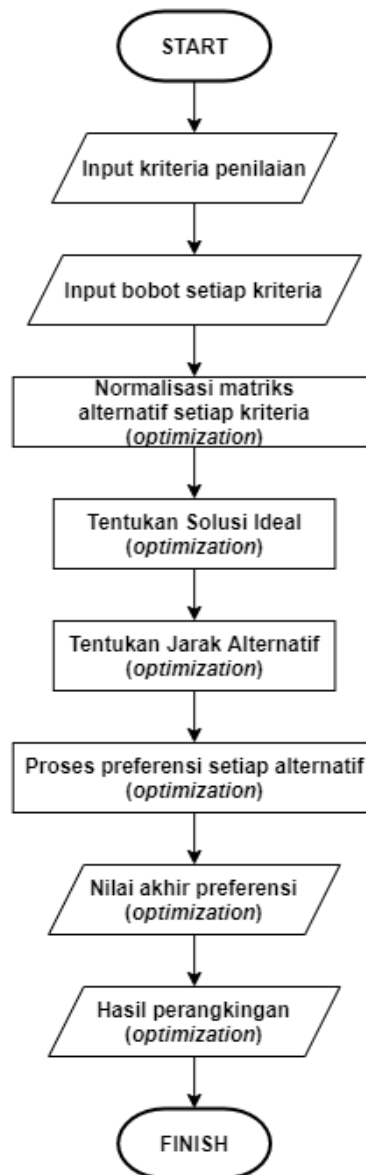


biaya ini bisa beberapa kali lipat besarnya. Jadi, semua DBMS memiliki modul yang memeriksa "semua" alternatif dan memilih rencana yang membutuhkan waktu paling sedikit. Modul ini disebut pengoptimal *query* (Ioannidis, 2004).

*Query* DSS menggunakan banyak input-output, pemrosesan dan sumber daya komunikasi dan dapat secara tiba-tiba menghentikan CPU atau bahkan server memori dari sistem database terdistribusi. Selain itu, waktu berjalan *query* DSS terdistribusi tidak dapat diperkirakan secara konvensional. *query* DSS bekerja pada relasi yang memiliki ukuran dalam mega byte, giga byte, atau bahkan lebih besar ukurannya (Sharma, 2016).

Pengoptimalan *query* akhirnya menjadi tantangan terbesar bagi para peneliti database. Optimasi *query* dalam sistem database terdistribusi telah mendapatkan perhatian yang cukup besar dalam beberapa tahun terakhir. Ini adalah proses untuk menentukan rencana eksekusi *query* terbaik dalam hal Total Biaya atau Waktu Respons. Rencana eksekusi *query* yang tidak optimal membutuhkan banyak biaya baik dalam hal penggunaan sumber daya sistem atau dalam hal waktu eksekusi. Dalam penelitian (Sharma, 2016), fokusnya adalah menganalisis kinerja pengoptimal *query* DSS terdistribusi stokastik yang berbeda. Hasilnya dibandingkan berdasarkan penggunaan sumber daya sistem dan waktu proses *query* DSS terdistribusi.

Dengan langkah-langkah yang sama, dilakukan optimasi pada setiap tahapan dalam metode TOPSIS sehingga akan terlihat seperti Gambar 2.2 berikut ;



**Gambar 2. 2** *Flowchart* Metode TOPSIS (*Query Optimization*)

Berdasarkan gambar 2.2 dimana terdapat *flowchart* untuk metode TOPSIS yang ditunjukkan secara berurutan. Namun yang membedakan *flowchart* pada gambar 2.2 dan gambar 2.1 terletak pada penambahan optimasi *query* pada setiap langkah-langkah TOPSIS.

Pada penelitian sebelumnya yang dilakukan oleh (Kuo, 2017) mengatakan bahwa, TOPSIS berupaya untuk memilih alternatif yang secara simultan memiliki jarak terdekat dari solusi ideal positif (PIS) dan jarak terjauh dari solusi ideal negatif (NIS) tidak peduli berapa bobot yang diberikan oleh pembuat keputusan untuk dua pemisahan ini, hasil pemeringkatan tidak akan berbeda seolah-olah dia tidak memiliki preferensi untuk kedua pemisahan ini. Oleh karena itu, penelitian ini berfokus pada memperlakukan pemisahan alternatif dari PIS dan NIS sebagai kriteria "*cost*" dan kriteria "*benefit*", masing-masing, mengurangi masalah MCDM (Pengambilan Keputusan Multi-Kriteria) asli menjadi masalah MCDM baru dengan dua kriteria ini hanya. Dengan mengusulkan  $w^-$  dan  $w^+$  masing-masing sebagai bobot kriteria "*cost*" dan kriteria "*benefit*", kemudian ditetapkan indeks peringkat baru.

Sedangkan pada penelitian ini, metode TOPSIS akan lebih difokuskan pada pengoptimasian *query* untuk setiap langkah-langkahnya. Hal ini dapat mempercepat pemrosesan *query* dan akan menghasilkan nilai *response times* yang lebih kecil dibandingkan dengan metode TOPSIS *non-optimization*. Sehingga diharapkan dapat membantu dalam menutupi kekurangan penelitian sebelumnya.

## **2.4 Response Time**

*Response time* yaitu respon waktu yang diberikan oleh *interface* ketika pengguna mengirim *request* ke komputer. Pada umumnya, *user* memiliki keinginan bahwa sistem yang dibangun bisa memberikan *response times* yang sesingkat-singkatnya. Akan tetapi *response times* yang ideal tidak dapat ditetapkan, karena terdapat beberapa komponen yang mempengaruhi, seperti jenis

interaksi yang diinginkan dan kelancaran pengguna dalam menggunakan sistem tersebut (Noviyanti, 2018).

*Response time* merupakan waktu yang ditempuh untuk memproses satu *statement query*. Disisi lain, *throughput* merupakan banyaknya *statement* yang bisa dijalankan dalam satu satuan waktu. Semakin besar *throughput* sebuah *database* serta semakin kecil *response time*, maka semakin baik pula performansinya (Martha, 2010).

## 2.5 Usability

*System Usability Scale* (SUS) adalah kumpulan pertanyaan yang biasa diterapkan dalam menghitung usability aplikasi komputer berdasarkan sudut pandang subyektif *user* (Brooke, 2013). SUS sering kali digunakan karena memiliki beberapa keunggulan, yaitu SUS dapat digunakan dengan mudah, karena hasilnya berupa skor 0–100. Selain itu, SUS tidak membutuhkan perhitungan yang rumit dan yang paling penting, SUS terbukti valid dan reliable, walau hanya dengan menggunakan ukuran sampel yang kecil (Nugroho, 2015).

Kuesioner SUS menggunakan lima poin skala Likert. Responden diminta agar memberikan penilaian “Sangat tidak setuju”, “Tidak setuju”, “Netral”, “Setuju”, dan “Sangat setuju” atas 10 daftar pernyataan SUS sesuai dengan penilaian subyektifnya. Nilai likert yang Daftar Pertanyaan SUS dapat dilihat pada tabel 2.1.

**Tabel 2. 1** Daftar Pertanyaan SUS

Kode	Pertanyaan
A1	Sistem sangat membantu user sehingga akan sering digunakan.

A2	Sistem mudah digunakan.
A3	Respon sistem baik dan tidak rumit.
A4	Fungsi terintegrasi dengan baik sehingga membutuhkan waktu yang cepat untuk mendapatkan hasil.
A5	Sistem mudah dipelajari dan digunakan.
A6	Bahasa dalam sistem konsisten mudah dimengerti
A7	Sistem sangat memudahkan dan membuat saya percaya diri dalam menggunakannya.
A8	Memberikan informasi lengkap tentang korban bencana sehingga saya tidak perlu bantuan teknisi dalam menggunakan sistem.
A9	Sistem menarik dan tidak banyak yang harus saya pelajari sebelum menggunakannya.
A10	Komposisi warna dan tampilan sudah baik sehingga membuat saya akan kembali menggunakan sistem ini.

Setiap item pernyataan memiliki skor kontribusi. Setiap skor kontribusi item akan berkisar antara 0 hingga 4. Untuk item 1,3,5,7, dan 9 skor kontribusinya adalah posisi skala dikurangi 1. Untuk item 2,4,6,8, dan 10, skor kontribusinya adalah 5 dikurangi posisi skala. Kalikan jumlah skor kontribusi dengan 2.5 untuk mendapatkan nilai keseluruhan system usability. Skor SUS berkisar dari 0 hingga 100 (Brooke, 1996). Berikut rumus perhitungan skor SUS:

$$\text{Skor SUS} = ((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1) + (5 - R8) + (R9 - 1) + (5 - R10)) * 2.5$$

Selanjutnya, Skor SUS keseluruhan diperoleh dari rata-rata skor SUS individual.

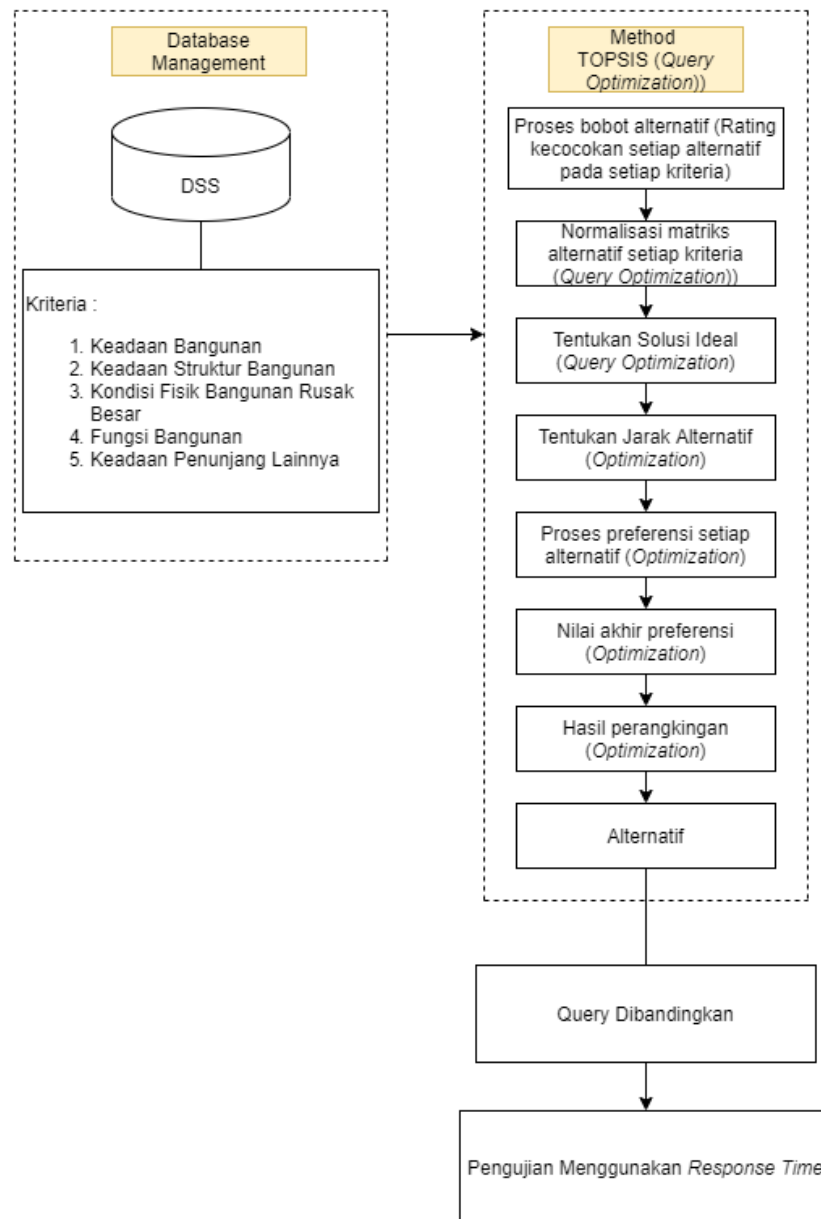
## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Desain Sistem**

Pada bagian ini akan dijelaskan tahapan dalam melakukan penelitian pada penelitian ini. Sistem yang akan dibangun dalam penelitian ini merupakan Sistem Pendukung Keputusan (SPK) dengan menggunakan metode TOPSIS. Metode TOPSIS akan diimplementasikan dalam bentuk *query optimization* menggunakan bahasa SQL (*Structure Query Language*). Untuk menguji metode TOPSIS pada *query optimization* menggunakan *respons time* untuk mengetahui waktu akses antara metode TOPSIS dengan *query optimization*. Terdapat 8 tahapan dalam proses mendapatkan alternatif yang sesuai dan sebelumnya sudah melalui tahap *optimization* terhadap *query Non-optimization* database metode topsis yang dapat dilihat pada gambar 3.1.

Berikut merupakan rancangan desain sistem yang akan digunakan dalam penelitian ini:



**Gambar 3. 1** Blok Diagram Sistem

Pada gambar 3.1, dijelaskan bahwa desain sistem terdiri dari 3 komponen yaitu *Database Management Method TOPSIS* biasa dan *Method TOPSIS Query Optimization*. Pada *Database Management* berisi kriteria yang akan digunakan untuk pengambilan keputusan menggunakan metode TOPSIS. Kemudian *Method TOPSIS Tradisional* menjelaskan tahapan atau langkah-langkah dalam metode

TOPSIS sedangkan pada *Method TOPSIS Query Optimization* menjelaskan tahapan langkah-langkah dalam metode TOPSIS dimana sudah dilakukan *query optimization* pada setiap tahapannya.

### 3.2 Data Collection

Pada penelitian ini, digunakan data yang sama dalam membandingkan *query optimization* dan *query non-optimization* yaitu, berupa data sekunder dimana data yang digunakan berasal dari data bencana kota Malang pada tahun 2019 yang merupakan bagian dari data Badan Nasional Penanggulangan Bencana(BNPB) di Indonesia.. Pada data tersebut terdapat beberapa kriteria yang dapat dijadikan acuan dalam penentuan kerusakan sektor pasca bencana. Dimana kriteria tersebut yang akan mendukung sistem yang akan dirancang untuk surveyor dalam menentukan kerusakan sektor pasca bencana. Data yang ditampilkan didapatkan oleh ahli dalam bidang terkait, di dalamnya terdapat data bencana alam baik berupa kerusakan rumah, instansi, jembatan, maupun masjid.

### 3.3 Identifikasi Perangkat Keras dan Perangkat Lunak yang Digunakan

**Tabel 3. 1** Spesifikasi Perangkat Lunak yang Digunakan

No	Software	Version
1	MongoDB	10.4.11
2	DBVisualizer	12.0.8

**Tabel 3. 2** Spesifikasi Perangkat Keras yang Digunakan

No	Item	Descriptions
1	Processor	AMD E2-7110 APU
2	Memory	8 GB
3	Operating System	Windows 10 Pro64-bit
4	Harddisk	500 GB



### 3.4 Alternatif dan Kriteria

Kriteria dan Alternatif yang menjadi acuan pada penelitian ini dapat dilihat pada tabel 3.3 dan tabel 3.4 berikut:

**Tabel 3. 3** Alternatif

<b>Alternatif</b>	<b>Nama Alternatif</b>
A1	Rusak Ringan
A2	Rusak Sedang
A3	Rusak Berat

Pada tabel 3.3 menunjukkan informasi Alternatif yang didapatkan dari data Pusat Vulkanologi dan Mitigasi Bencana Geologi (Badan Nasional Penanggulangan Bencana., 2019) di Provinsi Jawa Timur. Dalam penelitian ini penulis mengambil data pola kerusakan bencana di kota Malang Provinsi Jawa Timur. Alternatif tersebut antara lain Rusak Ringan, Rusak Sedang dan Rusak Berat.

**Tabel 3. 4** Kriteria

<b>Kriteria</b>	<b>Nama Kriteria</b>
C1	Kondisi Bangunan
C2	Kondisi Struktur Bangunan
C3	Kondisi Fisik Bangunan
C4	Fungsi Bangunan
C5	Kondisi Penunjang Lainnya

Pada tabel 3.4 memberikan informasi Kriteria yang didapatkan dari data pola kerusakan bencana di kota Malang. Kriteria tersebut antara lain Kondisi Bangunan, Kondisi Struktur Bangunan, Kondisi Fisik Bangunan, Fungsi Bangunan dan Kondisi Penunjang Lainnya.

### 3.5 Perhitungan Manual Metode TOPSIS

Hasil dari perhitungan Topsis nantinya akan digunakan sebagai data pola. Berikut adalah langkah Experiment untuk perhitungan manual Topsis.

#### 3.5.1 Membuat Matrik Keputusan

$$\begin{array}{l}
 L_1 \\
 L_2 \\
 \vdots \\
 L_n
 \end{array}
 \begin{bmatrix}
 C_1 & C_2 & \dots & C_n \\
 x_{11} & x_{12} & \dots & x_{1n} \\
 x_{21} & x_{22} & \dots & x_{2n} \\
 \vdots & \vdots & \dots & \vdots \\
 x_{m1} & x_{m2} & \dots & x_{mn}
 \end{bmatrix}$$

Keterangan :

L = Alternatif

C = Mengacu pada kriteria

X = Kriteria

**Tabel 3. 5** Perhitungan Matriks Keputusan

<b>Kepentingan</b>	3	5	5	4	3
<b>Kriteria</b>	<b>Kondisi Bangunan</b>	<b>Struktur Bangunan</b>	<b>Fisik Bangunan</b>	<b>Fungsi Bangunan</b>	<b>Kondisi Penunjang</b>
<b>A1</b>	4	3	4	5	4
<b>A2</b>	4	3	3	4	3
<b>A3</b>	3	4	2	2	2

#### 3.5.2 Matriks Ternormalisasi(R)

Setelah matriks keputusan didapatkan, berikutnya yaitu membentuk matriks keputusan ternormalisasi R yang berfungsi untuk mempersempit range data, dengan tujuan untuk memudahkan proses perhitungan TOPSIS dan penghematan dalam penggunaan memory. Berikut merupakan hasil dari Matriks Ternormalisasi:

**Tabel 3. 6** Matriks Ternormalisasi

<b>Ternormalisasi</b>	<b>Kondisi Bangunan</b>	<b>Struktur Bangunan</b>	<b>Fisik Bangunan</b>	<b>Fungsi Bangunan</b>	<b>Kondisi Penunjang</b>
<b>A1</b>	0.62469504	0.51449575	0.74278135	0.7453559	0.74278135
<b>A2</b>	0.62469504	0.51449575	0.55708601	0.5962847	0.55708601
<b>A3</b>	0.46852128	0.68599434	0.37139067	0.2981423	0.37139067

**3.5.3 Matriks Normalisasi Terbobot(Y)**

Setelah memperoleh matriks ternormalisasi, selanjutnya nilai pada matriks normalisasi dikalikan dengan nilai preferensi pada setiap kriteria:

**Tabel 3. 7** Matriks Normalisasi Terbobot

<b>Terbobot</b>	<b>Kondisi Bangunan</b>	<b>Struktur Bangunan</b>	<b>Fisik Bangunan</b>	<b>Fungsi Bangunan</b>	<b>Kondisi Penunjang</b>
<b>A1</b>	1.87408514	2.57247877	3.71390676	2.98142397	2.22834405
<b>A2</b>	1.87408514	2.57247877	2.78543007	2.38513917	1.67125804
<b>A3</b>	1.40556385	3.42997170	1.85695338	1.19256958	1.11417202

**3.5.4 Matriks Solusi Ideal Positif (A+)**

Solusi Ideal Positif (A+) merupakan nilai optimum maksimum (terbesar) dari suatu kriteria untuk beberapa nilai alternatif solusi dalam satu kriteria dengan hasil sebagai berikut:

**Tabel 3. 8** Matriks Solusi Ideal Positif (A+)

<b>Kondisi Bangunan</b>	<b>Struktur Bangunan</b>	<b>Fisik Bangunan</b>	<b>Fungsi Bangunan</b>	<b>Kondisi Penunjang</b>
1.874085143	3.429971703	3.713906764	2.98142397	2.228344058

**3.5.5 Matriks Solusi Ideal Positif (A-)**

Solusi Ideal Negatif (A-) merupakan nilai optimum minimum (terkecil) dari suatu kriteria untuk beberapa nilai alternatif solusi dalam satu kriteria dengan hasil sebagai berikut:

**Tabel 3. 9** Matriks Solusi Ideal Positif (A-)

<b>Kondisi bangunan</b>	<b>Struktur Bangunan</b>	<b>Fisik Bangunan</b>	<b>Fungsi Bangunan</b>	<b>Kondisi Penunjang</b>
1.405563857	2.572478777	1.856953382	1.192569588	1.114172029

**3.5.6 Matriks Jarak Solusi Ideal Positif (D+) Jarak Solusi Ideal Negatif (D-)**

Jarak Solusi Ideal adalah jarak antara nilai alternatif dengan nilai solusi ideal untuk setiap kriteria. Berikut ini merupakan nilai ideal positif dan ideal negatif:

**Tabel 3. 10** Matriks D+ dan D

<b>D+</b>	<b>D-</b>
0.8574929	2.84766
1.5044147	1.6775423
2.84766	0.8574929

**3.5.7 Perangkingan**

Dengan demikian bisa diurutkan hasilnya dari yang terbesar hingga yang terkecil, dimana nilai preferensi dari alternatif yang terbesar ialah alternatif terbaik dari data yang ada serta adalah alternatif yang terpilih, sedangkan alternatif dengan nilai optimasi terendah adalah yang terburuk dari data yang terdapat. pada urutan dari yang terbesar hingga yang terkecil, diperoleh :

**Tabel 3. 11** Perangkingan Beserta Alternatif

<b>V</b>	<b>Hasil</b>	<b>Alternatif</b>
0.7685675	A1	Rusak Ringan
0.5272046	A2	Rusak Sedang
0.2314325	A3	Rusak Berat

Dari hasil perhitungan secara manual yang sudah dilakukan, alternatif dengan kode A1 yaitu Rusak Ringan memiliki nilai preferensi tertinggi dengan

nilai preferensi 0.7685675, lebih tinggi dibandingkan dengan alternatif A2 dengan nilai 0.5272046 dan alternatif A3 dengan nilai 0.2314325.

### 3.6 Perbandingan *Query optimization* dan *Non-Optimization*

Perkiraan pemrosesan *query* baru-baru ini muncul sebagai solusi yang layak untuk menangani sejumlah besar data, kompleksitas *query* yang tinggi, dan persyaratan *response time* yang semakin ketat yang menjadi ciri aplikasi sistem pendukung keputusan (DSS) saat ini. Biasanya, pengguna DSS memiliki sikap yang sangat kompleks *query* ke sistem manajemen basis data (DBMS) yang mendasari yang memerlukan operasi kompleks melalui Gigabyte atau Terabyte dari data penduduk disk dan, dengan demikian, membutuhkan waktu yang sangat lama untuk dieksekusi hingga selesai dan menghasilkan jawaban yang tepat (Chakrabarti, 2001).

Berikut beberapa contoh perbedaan untuk *query* biasa dan *query* yang sudah dioptimasi (Kumari, 2012):

#### 1. Penggunaan SELECT\*

*Query* Asli :

```
SELECT * FROM table_name;
```

*Query* Optimasi :

```
SELECT col_1, col_2, col_3, col_4,  
subject FROM table_name;
```

SELECT\* digunakan untuk melakukan *query* semua field yang terdapat pada sebuah table, tetapi jika hanya ingin memproses field tertentu, maka sebaiknya ditulis field yang ingin diakses saja, sehingga *query*

menjadi SELECT field1, field2, field3 dan seterusnya. Hal ini akan mengurangi beban lalu lintas jaringan dan lock pada table, terutama jika table tersebut memiliki banyak field dan berukuran besar.

## 2. Penggunaan HAVING

*Query Asli :*

```
SELECT Col_1, count (Col_1)
FROM table_name GROUP BY Col_1
HAVING Col_1!= 'testvalue1'
AND Col_1!= 'testvalue2';
```

*Query Optimasi :*

```
SELECT Col_1, count (Col_1)
FROM table_name WHERE col_1!= 'testvalue1'
AND col_1!= 'testvalue1' GROUP BY col_1;
```

Klausa HAVING digunakan untuk memfilter baris setelah semua baris dipilih dan digunakan seperti filter. Cobalah untuk tidak menggunakan klausa HAVING untuk tujuan lain.

## 3. Penggunaan UNION ALL

*Query Asli :*

```
SELECT id, col1, col2 FROM table1
UNION SELECT id, col1 FROM table2;
```

*Query Optimasi :*

```
SELECT id, col1 FROM table1
```

```
UNION ALL SELECT id,  
col1 FROM table2;
```

Pernyataan UNION ALL lebih cepat daripada UNION, karena pernyataan UNION ALL tidak mempertimbangkan duplikat, dan pernyataan UNION mencari duplikat dalam tabel saat pemilihan baris, apakah ada atau tidak.

#### 4. Penggunaan EXIST

*Query Asli :*

```
SELECT * FROM table1 t1  
WHERE t1.col_id IN (SELECT t2.col_id  
FROM table2 t2);
```

*Query Optimasi :*

```
SELECT * FROM table1 t1 WHERE  
EXISTS (SELECT * FROM table2 t2  
WHERE t2.col_id = t1.col_id)
```

EXISTS efisien ketika sebagian besar kriteria filter untuk pemilihan ada di kueri utama pernyataan SQL.

### 3.7 Rencana Pengujian

Dalam melakukan pengujian, guna mendapatkan hasil yang baik, berikut faktor yang akan digunakan untuk melihat efektifitas *query optimization* yang diterapkan pada penelitian ini.

#### 3.7.1 Membandingkan Dua Jenis *Query* (*Query Non-Optimization* dan *Query Optimization*)

Pada penelitian ini akan disajikan tabel perbedaan antar penggunaan *query* TOPSIS tradisional dan *query* yang sudah dioptimasi. Akan ditunjukkan letak

perbedaan antara kedua *query* sehingga dapat dijadikan acuan ketika ingin melakukan optimasi pada *query* lainnya.

### 3.7.2 Pengujian Menggunakan Response Time

Setelah menambahkan data pada database, pengujian terhadap *query optimization* langsung dapat dilakukan dengan melihat hasil *response time* yang muncul saat melakukan *request query*.

Dalam percobaan yang dilakukan dalam penelitian ilmiah ini memiliki 6 skenario sesuai dengan tahapan pada metode TOPSIS yang dijalankan, yaitu normalisasi matriks, menentukan solusi ideal, menentukan jarak alternatif, proses preferensi, nilai akhir preferensi, dan hasil perangkingan. Setiap skenario dilakukan dengan mengeksekusi jumlah data yang berbeda yaitu 114, 228, dan 342 data. *Query* dieksekusi 3 kali dan mengambil rata-rata untuk mendapatkan hasil akhir.

Untuk menguji *response time* pada database, penelitian ini menggunakan DbVisualizer 12.0.8 untuk melihat hasil *response time* yang didapat saat mengeksekusi *query*. Pertama, Dbvisualizer harus terhubung dengan database yang digunakan. Ia bekerja dengan menyesuaikan port, nama database, user id dan password database. Setelah berhasil terhubung, maka dapat dilakukan *request* terhadap tabel yang diinginkan.

Untuk setiap satu kali percobaan, hasil *response time query* ditampilkan dalam grafik. Grafik tersebut nantinya akan menggambarkan perbandingan trendline untuk setiap percobaan.

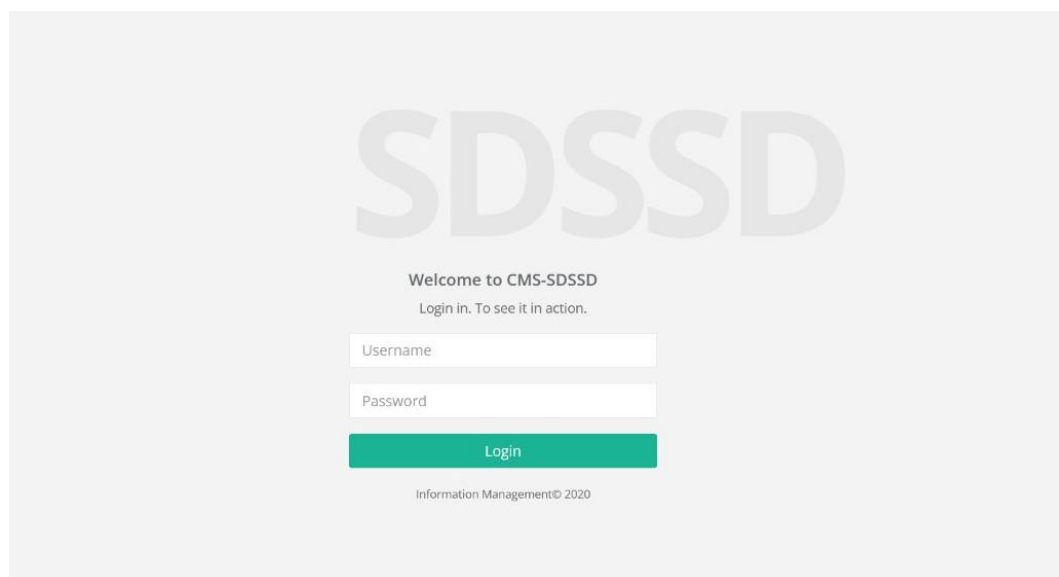


### 3.8 Implementasi Sistem

Dalam aplikasi ini, telah dibagi 2 *level*, yaitu *level* Super Admin dan *level* Admin. Setiap *level* memiliki fungsi dan tampilan yang berbeda, namun tetap memiliki keterkaitan antara satu dan yang lainnya. Berikut *interface* untuk aplikasi pada penelitian ini :

#### 1. Halaman Login

Halaman yang akan muncul pertama kali saat aplikasi diakses oleh *admin* maupun *super admin* disebut dengan halaman *Login*. Agar dapat membuka halaman selanjutnya, dibutuhkan akun yang sudah terdaftar pada database. Akun tersebut berisikan *username* dan *password* yang akan digunakan untuk mengkonfirmasi *login*. Halaman login dapat dilihat pada Gambar 3.2.



**Gambar 3. 2** Halaman Login

#### 2. Halaman Super Admin

Halaman yang dapat diakses oleh *Super Admin* sebagai berikut :

##### a. Halaman Data Kriteria

Pada halaman data kriteria, super admin dapat menambahkan ataupun menghapus data kriteria seperti pada gambar 3.3.

#	Nama Kriteria	update	delete
1	Keadaan Bangunan	update	delete
2	Keadaan Struktur Bangunan	update	delete
3	Kondisi Fisik Bangunan Rusak Sebesar	update	delete
4	Fungsi Bangunan	update	delete
5	Keadaan Penunjang Lainnya	update	delete

**Gambar 3. 3** Halaman Super Admin Data Kriteria

#### b. Halaman Data Alternatif

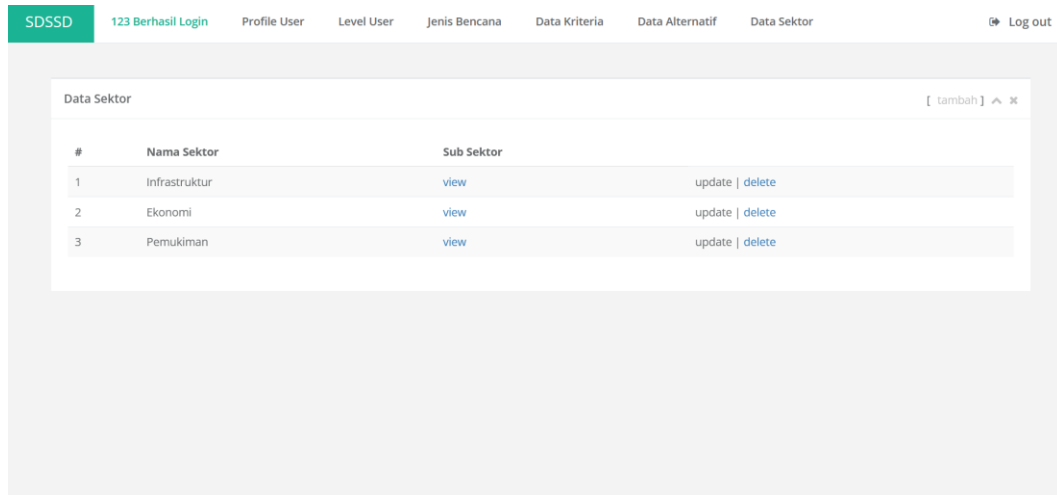
Pada halaman data alternatif, super admin dapat menambahkan ataupun menghapus data alternatif seperti pada gambar 3.4.

#	Nama Alternatif	update	delete
1	Rusak Ringan	update	delete
2	Rusak Sedang	update	delete
3	Rusak Berat	update	delete

**Gambar 3. 4** Halaman Super Admin Data Alternatif

#### c. Halaman Data Sektor

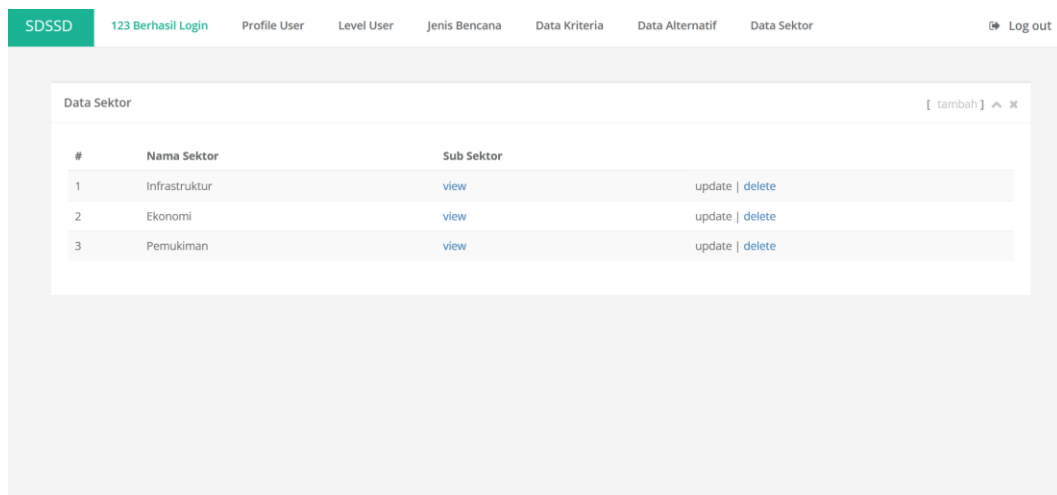
Pada halaman data sektor, super admin dapat menambahkan ataupun menghapus data sektor seperti pada gambar 3.5.



**Gambar 3. 5** Halaman Super Admin Data Sektor

#### d. Halaman Jenis Bencana

Pada halaman jenis bencana, super admin dapat menambahkan ataupun menghapus jenis bencana seperti pada gambar 3.6.



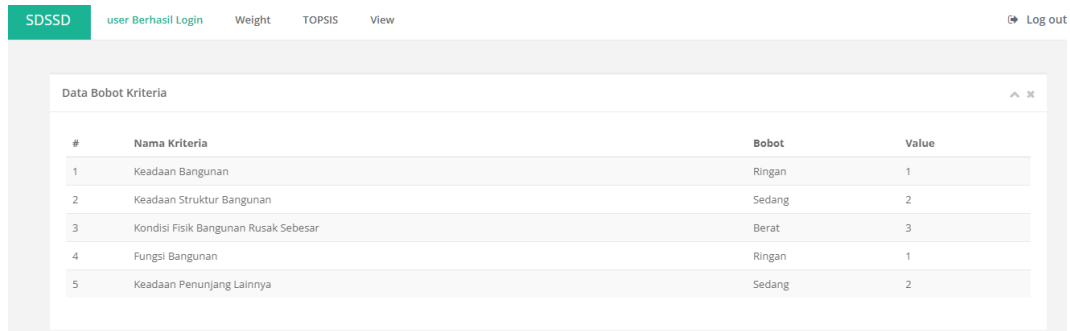
**Gambar 3. 6** Halaman Super Admin Jenis Bencana

### 3. Level Admin

Halaman yang dapat diakses oleh *Super Admin* sebagai berikut :

a. Halaman Weight

Pada halaman *weight* atau disebut juga data bobot, ditampilkan data bobot setiap kriteria seperti yang terdapat pada gambar 3.7.



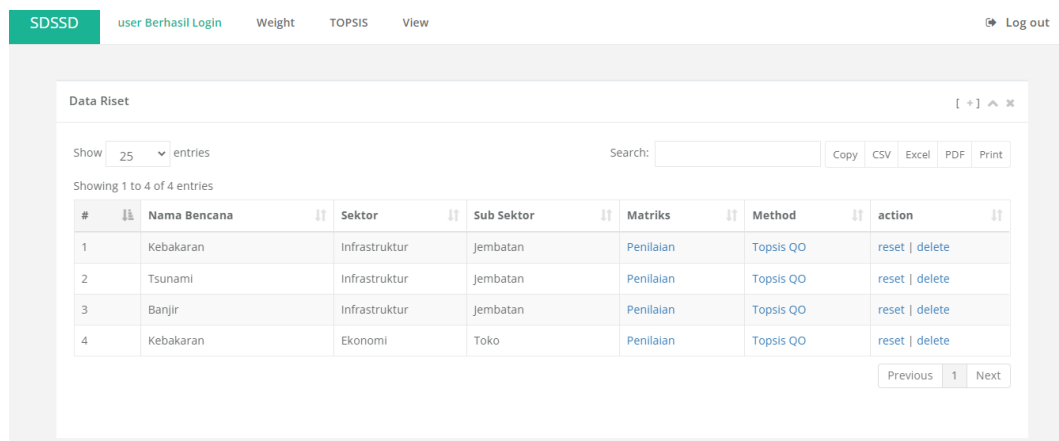
The screenshot shows the 'Data Bobot Kriteria' table in the SDSSD application. The table has four columns: '#', 'Nama Kriteria', 'Bobot', and 'Value'. It contains five rows of data representing different building criteria and their assigned weights.

#	Nama Kriteria	Bobot	Value
1	Keadaan Bangunan	Ringan	1
2	Keadaan Struktur Bangunan	Sedang	2
3	Kondisi Fisik Bangunan Rusak Sebesar	Berat	3
4	Fungsi Bangunan	Ringan	1
5	Keadaan Penunjang Lainnya	Sedang	2

**Gambar 3. 7** Halaman Admin Weight

b. Halaman Topsis

Pada halaman TOPSIS, admin dapat melakukan tambah dan hapus data seperti pada gambar 3.8. Selain itu, admin dapat melihat hasil dari perhitungan metode TOPSIS pada gambar 3.9 dan hasil dari penilaian seperti pada gambar 3.10.



The screenshot shows the 'Data Riset' table in the SDSSD application. The table has columns for '#', 'Nama Bencana', 'Sektor', 'Sub Sektor', 'Matriks', 'Method', and 'action'. It contains four rows of data representing different disaster types and their associated sectors and sub-sectors.

#	Nama Bencana	Sektor	Sub Sektor	Matriks	Method	action
1	Kebakaran	Infrastruktur	Jembatan	Penilaian	Topsis QO	reset   delete
2	Tsunami	Infrastruktur	Jembatan	Penilaian	Topsis QO	reset   delete
3	Banjir	Infrastruktur	Jembatan	Penilaian	Topsis QO	reset   delete
4	Kebakaran	Ekonomi	Toko	Penilaian	Topsis QO	reset   delete

**Gambar 3. 8** Halaman Admin TOPSIS

#	Kriteria	Pembagi
1	Keadaan Bangunan	4.123105625617661
2	Keadaan Struktur Bangunan	3.7416573867739413
3	Kondisi Fisik Bangunan Rusak Sebesar	4.69041575982343
4	Fungsi Bangunan	3.7416573867739413
5	Keadaan Penunjang Lainnya	3

#	Kriteria	Alternatif	Penilaian	Normalisasi	Terbobot
1	Keadaan Bangunan	Rusak Ringan	Sedang	0.8164965809277261	0.8164965809277261
2	Keadaan Bangunan	Rusak Berat	Sedang	0.8164965809277261	0.8164965809277261
3	Keadaan Bangunan	Rusak Sedang	Berat	1.2247448713915892	1.2247448713915892
4	Keadaan Struktur Bangunan	Rusak Berat	Ringan	0.4082482904638631	0.8164965809277261
5	Keadaan Struktur Bangunan	Rusak Sedang	Sedang	0.8164965809277261	1.6329931618554523
6	Keadaan Struktur Bangunan	Rusak Ringan	Berat	1.2247448713915892	2.4494897427831783
7	Kondisi Fisik Bangunan Rusak Sebesar	Rusak Sedang	Sedang	0.6030226891555273	1.8090680674665818
8	Kondisi Fisik Bangunan Rusak Sebesar	Rusak Berat	Berat	0.9045340337332909	2.7136021011998728
9	Kondisi Fisik Bangunan Rusak Sebesar	Rusak Ringan	Berat	0.9045340337332909	2.7136021011998728
10	Fungsi Bangunan	Rusak Ringan	Berat	1.7320508075688774	1.7320508075688774

**Gambar 3. 9** Halaman Admin Topsis QO

Alternatif	D+	D-
Rusak Ringan	0.7977240352174658	2.6363504649179674
Rusak Sedang	2.1444226519772673	1.5119842117841351
Rusak Berat	2.184220093735117	2.3605491974049135

Alternatif	TOPSIS
Rusak Ringan	0.767703340394623
Rusak Sedang	0.41351640233733
Rusak Berat	0.5193991259373218

**Gambar 3. 10** Halaman Admin Hasil Alternatif

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

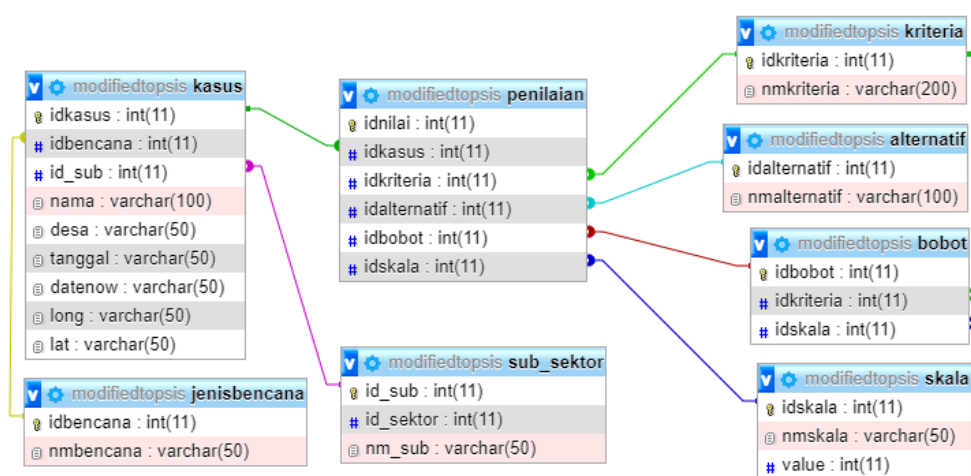
Pada bab ini akan dijelaskan mengenai hasil dan pembahasan dari upaya dalam mengoptimasi *query* TOPSIS untuk

#### **4.1 Implementasi *Query Optimization***

Pada tabel 3.2 memberikan informasi Kriteria yang didapatkan dari data pola kerusakan bencana di kota Blitar. Kriteria tersebut antara lain Kondisi Struktur Bangunan, Kondisi Bangunan, Fungsi Bangunan, Kondisi Fisik Bangunan dan Kondisi Penunjang Lainnya.

Setelah itu data disimpan pada database Maria DB untuk di proses menjadi metode TOPSIS. Untuk proses implementasi metode TOPSIS pada database MariaDB menggunakan function view yang ada pada library database (MariaDB). Setiap langkah yang terdapat pada tabel 2 diatas di implementasikan dengan perintah *query* kemudian hasil dari *query* tersebut di simpan kedalam view. View merupakan suatu function yang berfungsi untuk menyimpan *query*, *query* tersebut berisi data-data yang akan digunakan pada langkah *query* metode TOPSIS selanjutnya. Hasil dari view hampir sama dengan table di database tetapi view tidak bisa melakukan CRUD (Create, Insert, Update, Delete) data yang tersimpan di view tersebut. Ada perubahan data yang di view jika ada perubahan data pada tabel yang bersangkutan dengan view tersebut. Untuk memanggil view dengan menggunakan perintah *query* 'SELECT' seperti pada saat memanggil data pada table di dalam database MariaDB.

Pertama yang dilakukan adalah membuat rancangan databasenya agar metode TOPSIS bisa diimplementasikan pada *query* dengan baik tanpa ada redundance data pada setiap langkahnya. Database yang baik merupakan database yang memiliki tingkat redundance data sangat kecil. Berikut gambar 1 merupakan desain database hasil dari rancangan database sebelum metode TOPSIS diimplementasikan pada database MariaDB.



**Gambar 4. 1** Desain Database

Gambar design database 1 merupakan dasar dari metode TOPSIS yang diimplementasikan menggunakan bahasa *query*. Hasil implementasi *query* untuk metode TOPSIS akan berbentuk SQL seperti tabel 4. Pada tabel 4 terdapat 5 langkah metode TOPSIS, masing-masing langkah diimplementasikan dengan menggunakan *query non-optimization* dan *query optimization* untuk mencari *response time*.

**Tabel 4. 1** Perbandingan *Query Optimization* dan Non-Optimization

<b>TOPSIS Langkah ke-</b>	<i>Query Non-Optimization</i>	<i>Query Optimization</i>
1	<pre>select `penilaian`.`idkasus` AS `idkasus`,`kriteria`.`idkriteria` AS `idkriteria`,`penilaian`.`idalternati f` AS `idalternatif`,sqrt(sum(pow(`skala `.`value`,2))) AS `bagi` from ((`penilaian` join `skala`) join `kriteria`) where `skala`.`idskala` = `penilaian`.`idskala` and `kriteria`.`idkriteria` = `penilaian`.`idkriteria` group by `kriteria`.`idkriteria`,`penilaian`.`i dkasus`</pre>	<pre>SELECT `penilaian`.`idkasus` AS `idkasus`,`kriteria`.`idkriteria` AS `idkriteria`,`penilaian`.`idalternatif ` AS `idalternatif`,SQRT(SUM(POW(` skala`.`value`,2))) AS `bagi` FROM ((penilaian JOIN skala) JOIN kriteria) WHERE `kriteria`.`idkriteria` = penilaian.idkriteria AND `skala`.`idskala` = penilaian.idskala GROUP BY `kriteria`.`idkriteria`,`penilaian`.`i dkasus`</pre>
2	<pre>select `penilaian`.`idnilai` AS `idnilai`,`penilaian`.`idkasus` AS `idkasus`,`penilaian`.`idkriteria` AS `idkriteria`,`penilaian`.`idalternati f` AS</pre>	<pre>SELECT `penilaian`.`idnilai` AS `idnilai`,`penilaian`.`idkasus` AS `idkasus`,`penilaian`.`idkriteria` AS `idkriteria`,`penilaian`.`idalternatif ` AS</pre>



	<pre> `idalternatif`,`penilaian`.`idbobot` AS `idbobot`,`penilaian`.`idskala` AS `idskala`,`skala`.`value` / `1pembagi`.`bagi` AS `normalisasi` from ((`penilaian` join `1pembagi`) join `skala`) where `penilaian`.`idskala` = `skala`.`idskala` and `1pembagi`.`idkriteria` = `penilaian`.`idkriteria` group by `penilaian`.`idnilai`,`penilaian`.`id kasus` </pre>	<pre> `idalternatif`,`penilaian`.`idbobot` AS `idbobot`,`penilaian`.`idskala` AS `idskala`,`skala`.`value` / `1pembagi`.`bagi` AS `normalisasi` FROM ((`penilaian` JOIN `1pembagi`) JOIN `skala`) WHERE `1pembagi`.`idkriteria` = penilaian.idkriteria AND `penilaian`.`idskala` = skala.idskala GROUP BY `penilaian`.`idnilai`,`penilaian`.`id kasus` </pre>
3	<pre> select `2normalisasi`.`idnilai` AS `idnilai`,`2normalisasi`.`idkasus` AS `idkasus`,`2normalisasi`.`idkriteri a` AS `idkriteria`,`2normalisasi`.`idalter natif` AS `idalternatif`,`2normalisasi`.`idbo bot` AS `idbobot`,`2normalisasi`.`idskala` AS </pre>	<pre> SELECT `2normalisasi`.`idnilai` AS `idnilai`,`2normalisasi`.`idkasus` AS `idkasus`,`2normalisasi`.`idkriteri a` AS `idkriteria`,`2normalisasi`.`idalter natif` `AS `idalternatif`,`2normalisasi`.`idbo bot` AS </pre>

	<pre> `idskala`,`2normalisasi`.`normalis asi` AS `normalisasi`,`skala`.`value` AS `value`,`skala`.`value` * `2normalisasi`.`normalisasi` AS `terbobot` from ((`2normalisasi` join `bobot`) join `skala`) where `bobot`.`idskala` = `skala`.`idskala` and `bobot`.`idkriteria` = `2normalisasi`.`idkriteria` group by `2normalisasi`.`idnilai`,`2normalis asi`.`idkasus` </pre>	<pre> `idbobot`,`2normalisasi`.`idskala` AS `idskala`,`2normalisasi`.`normalis asi` AS `normalisasi`,`skala`.`value` AS `value`,`skala`.`value` * `2normalisasi`.`normalisasi` AS `terbobot` FROM ((`2normalisasi` JOIN `bobot`) JOIN `skala`) WHERE bobot.`idskala` = skala.`idskala` AND bobot.`idkriteria` = 2normalisasi.`idkriteria` GROUP BY `2normalisasi`.`idnilai`,`2normalis asi`.`idkasus` </pre>
4	<pre> select `3terbobot`.`idkasus` AS `idkasus`,`3terbobot`.`idkriteria` AS `idkriteria`,max(`3terbobot`.`terbo bot`) AS `maximum`,min(`3terbobot`.`terb obot`) AS `minimum` from </pre>	<pre> SELECT 3terbobot.`idkasus` AS `idkasus`, 3terbobot.`idkriteria` AS `idkriteria`,MAX(3terbobot.`terbo bot`) AS `maximum`,MIN(3terbobot.`terbo bot`) AS `minimum` FROM </pre>

	<pre> `3terbobot` group by `3terbobot`.`idkriteria`,`3terbobot` `.`idkasus` </pre>	<pre> 3terbobot GROUP BY `3terbobot`.`idkriteria`,`3terbobot` `.`idkasus` </pre>
5	<pre> select `3terbobot`.`idkasus` AS `idkasus`,`3terbobot`.`idalternatif` AS `idalternatif`,sqrt(sum(pow(`4ama x_amin`.`maximum` - `3terbobot`.`terbobot`,2))) AS `dplus`,sqrt(sum(pow(`4amax_am in`.`minimum` - `3terbobot`.`terbobot`,2))) AS `dmin` from (`3terbobot` join `4amax_amin`) where `3terbobot`.`idkriteria` = `4amax_amin`.`idkriteria` group by `3terbobot`.`idalternatif`,`3terbob ot`.`idkasus` </pre>	<pre> SELECT 3terbobot.`idkasus` AS `idkasus`, 3terbobot.`idalternatif` AS `idalternatif`,SQRT(SUM(POW(4 amax_amin.`maximum` - 3terbobot.`terbobot`,2))) AS `dplus`,SQRT(SUM(POW(4amax _amin.`minimum` - 3terbobot.`terbobot`,2))) AS `dmin` FROM (3terbobot JOIN 4amax_amin) WHERE 3terbobot.`idkriteria` = 4amax_amin.`idkriteria` GROUP BY `3terbobot`.`idalternatif`,`3terbob ot`.`idkasus` </pre>
6	<pre> select `5nilaid`.`idkasus` AS `idkasus`,`5nilaid`.`idalternatif` AS `idalternatif`,`5nilaid`.`dplus` AS `dplus`,`5nilaid`.`dmin` AS </pre>	<pre> SELECT 5nilaid.`idkasus` AS `idkasus`, 5nilaid.`idalternatif` AS `idalternatif`, 5nilaid.`dplus` AS `dplus`, 5nilaid.`dmin` AS `dmin`, </pre>

<pre> `dmin`,`5nilaid`.`dmin` / (^5nilaid`.`dplus` + `5nilaid`.`dmin`) AS `konvensional`,sqrt(pow(`5nilaid`.` dplus` - `5nilaid`.`dmin`,2)) AS `modified` from `5nilaid` group by `5nilaid`.`idkasus`,`5nilaid`.`idalte rnatif` </pre>	<pre> 5nilaid.`dmin` / (5nilaid.`dplus` + 5nilaid.`dmin`) AS `konvensional` FROM 5nilaid GROUP BY 5nilaid.`idkasus`, 5nilaid.`idalternatif` </pre>
--	--

Pada langkah TOPSI yang pertama dilakukan *optimization* pada beberapa bagian seperti kondisi WHERE dimana setelah tanda '=', kutip satu dihilangkan. Kemudian pada nama tabel usai keyword sql FROM tidak perlu menggunakan kutip satu. Selanjutnya, konsisten menggunakan huruf kapital pada keyword SQL. Terakhir menempatkan kondisi yang lebih sederhana untuk diproses di sebelah kiri keyword AND.

Pada langkah TOPSIS yang kedua dilakukan *optimization* pada beberapa bagian yaitu kondisi WHERE dimana setelah tanda '=', kutip satu dihilangkan. Lalu nama tabel usai keyword sql FROM tidak perlu menggunakan kutip satu. Selanjutnya, konsisten menggunakan huruf kapital pada keyword SQL. Kemudian, menempatkan kondisi yang lebih sederhana untuk diproses di sebelah kiri keyword AND. Pada dasarnya, bagian yang dioptimasi pada langkah pertama dan kedua tidak jauh berbeda.

Pada Langkah TOPSIS yang ketiga dilakukan *optimization* pada beberapa bagian seperti nama tabel sebelum dan sesudah keyword sql FROM tidak perlu menggunakan kutip satu dan konsisten menggunakan huruf kapital pada keyword SQL.

Pada Langkah TOPSIS yang keempat dilakukan *optimization* pada beberapa bagian seperti Pemanggilan nama view(3terbobot) tidak perlu menggunakan kutip satu dan konsisten menggunakan huruf kapital pada keyword SQL.

1. Pemanggilan nama view(3terbobot) tidak perlu menggunakan kutip satu
2. Konsisten menggunakan huruf kapital pada keyword SQL

#### **4.2 Uji Coba *Query Optimization***

Uji coba dilakukan menggunakan 3 data kerugian sektor pasca bencana alam dengan jumlah yang berbeda-beda. Setiap data akan di lakukan percobaan sebanyak 3 kali percobaan untuk mendapatkan waktu respon time. *Response time* yang dihasilkan dari 3 kali percobaan tersebut akan di cari rata-ratanya (*average*) untuk mendapatkan nilai *response time* pada setiap langkah metode TOPSIS pada setiap data uji coba. Tabel 4.2 berikut merupakan rincian atau detail jumlah data yang di uji coba:

**Tabel 4. 2** Jumlah data untuk uji coba

<b>Ujian Ke-</b>	<b>Jumlah Data</b>
1	114
2	228
3	342

Untuk menguji response time pada database, penelitian ini menggunakan DbVisualizer 12.0.8 untuk melihat hasil response time yang didapat saat

mengeksekusi *query*. Pertama, Dbvisualizer harus terhubung dengan database yang digunakan. Ia bekerja dengan menyesuaikan port, nama database, user id dan password database. Setelah berhasil terhubung, ia dapat meminta tabel. Misalnya pada gambar 4.2 dan gambar 4.3 terdapat tabel 2normalisasi dan tabel 2normalisasi\_optimasi. Kemudian hasil waktu respons ditampilkan di log di mana ada waktu exec dan fetch untuk hasilnya.

Time	Status	Command	Exec	Fetch	Rows	Message	SQL/Command
16:19:33	STARTED					Executing for: 'response time' [MariaDB], Database: topsis	
16:19:33	SUCCESS	SELECT*FROM	0.031	0.016	317	Result set fetched	SELECT*FROM 2normalisasi
16:19:34	FINISH...		0.031	0.016	317	Success: 1	

**Gambar 4. 2** *Response Time* Tabel 2normalisasi

Time	Status	Command	Exec	Fetch	Rows	Message	SQL/Command
16:20:46	STARTED					Executing for: 'response time' [MariaDB], Database: topsis	
16:20:46	SUCCESS	SELECT*FROM	0.030	0.001	317	Result set fetched	SELECT*FROM 2normalisasi_optimasi
16:20:47	FINISH...		0.030	0.001	317	Success: 1	

**Gambar 4. 3** *Response Time* Tabel 2normalisasi\_optimasi

#### 4.2.1 Uji Coba Ke-1

Uji coba ke-1 menggunakan data sebanyak 114 data yang sudah di inputkan pada database MariaDB. Pada tabel 4.3 terlihat bahwa uji coba ke-1 dilakukan 3 kali percobaan yang menghasilkan *response time* berbeda-beda pada setiap percobaan.

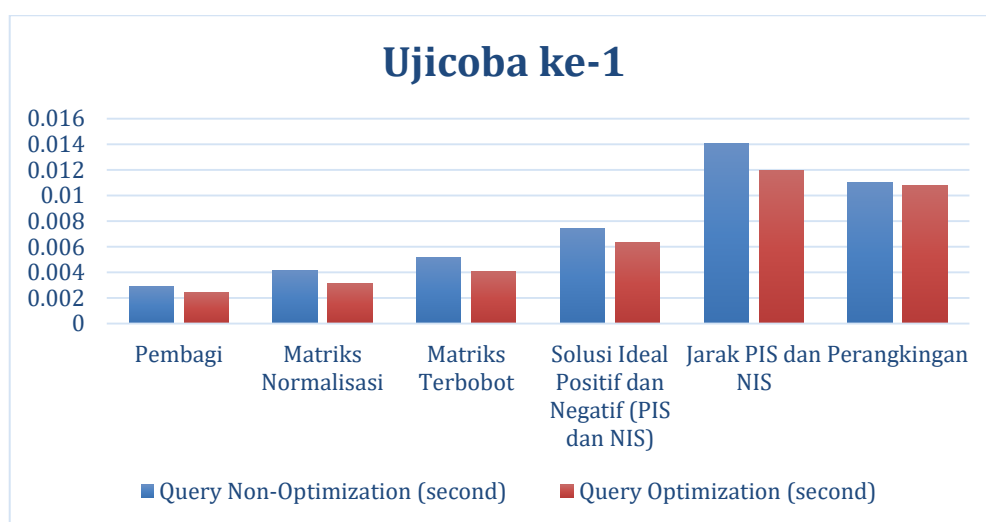
**Tabel 4. 3** Tabel uji coba ke-1

Langkah TOPSIS	Percobaan						Average	
	1		2		3		QNO (Mili second)	QO (Mili second)
	QNO (Mili second)	QO (Mili second)	QNO (Mili second)	QO (Mili second)	QNO (Mili second)	QO (Mili second)		
1.	4.6	3.6	2	1.9	2.1	1.9	2.7	2.466
2.	5.1	3.3	3.6	3.1	3.7	3	4.133	3.133
3.	5.2	4.1	6	4.2	4.4	3.8	5.2	4.033
4.	7.4	6.4	8.3	6.2	6.5	6.4	7.4	6.333
5.	14.4	12.6	15.9	12.8	11.8	10.5	14.033	11.966
6.	10.8	10.7	11.2	10.9	11	10.8	11	10.8

QNO = *Query Non-Optimization*

QO = *Query Optimization*

Dari 3 kali percobaan pada uji coba ke-1 yang berjumlah 114 data diambil rata-rata (*average*) *response time* pada masing-masing langkah metode TOPSIS yang menggunakan QNO (*Query Non-Optimization*) dan QO (*Query Optimization*). Hasil dari *average* masing-masing langkah metode TOPSIS berdasarkan QNO (*Query Non-Optimization*) dan QO (*Query Optimization*) dapat dilihat pada gambar 4.4 berikut.



**Gambar 4. 4** Grafik Uji Coba ke-1

Pada gambar 4.4 dapat dilihat bahwa pada uji coba ke-1 dengan menggunakan QO (*Query Optimization*) dapat mempercepat waktu akses (*Response time*) pada proses setiap langkah metode TOPSIS.



#### 4.2.2 Uji Coba ke-2

Uji coba ke-2 menggunakan data sebanyak 228 data yang sudah di inputkan pada database MariaDB. Pada tabel 4.4 terlihat bahwa uji coba ke-2 dilakukan 3 kali percobaan yang menghasilkan *response time* berbeda-beda pada setiap percobaan.

**Tabel 4. 4** Tabel Uji coba ke-2

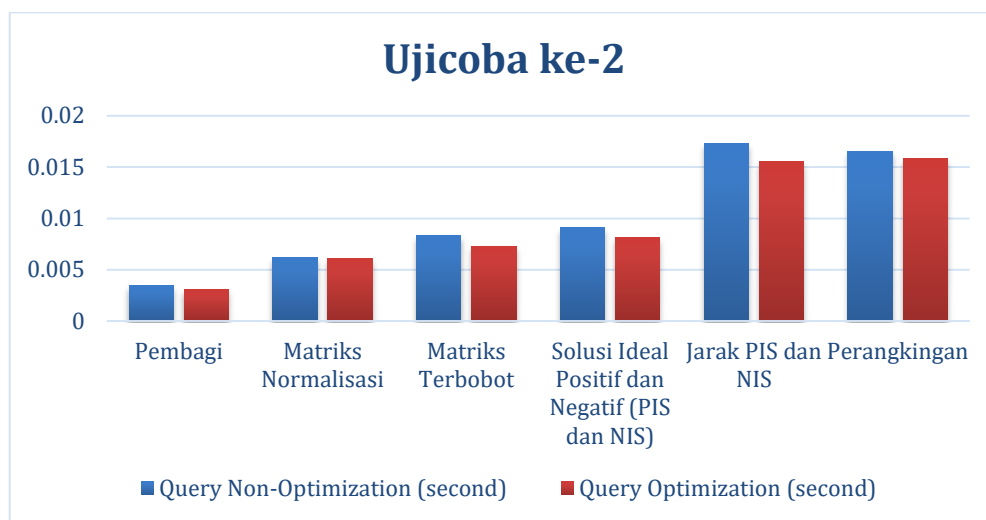
Langkah TOPSIS	Percobaan						Average	
	1		2		3		QNO (Mili second)	QO (Mili second)
	QNO (Mili second)	QO (Mili second)	QNO (Mili second)	QO (Mili second)	QNO (Mili second)	QO (Mili second)		
1.	3.3	3	4.2	3.2	2.9	3	3.466	3.066
2.	6.9	5.8	5.9	6.6	5.8	5.8	6.2	6.066
3.	7.6	7.2	10	7.4	7.4	7.3	8.333	7.3
4.	11.1	8.1	7.9	8.1	8.3	8.1	9.1	8.1
5.	15.7	15.5	20.6	15.9	15.7	15.1	17.333	15.5
6.	16	15.5	17.7	16	16	15.9	16.566	15.8

QNO = *Query Non-Optimization*

QO = *Query Optimization*

Dari 3 kali percobaan pada uji coba ke-2 yang berjumlah 228 data diambil rata-rata (*average response time*) pada masing-masing langkah metode TOPSIS yang menggunakan QNO (*Query Non-Optimization*) dan QO (*Query Optimization*). Hasil dari

average masing-masing langkah metode TOPSIS berdasarkan QNO (*Query Non-Optimization*) dan QO (*Query Optimization*) dapat dilihat pada gambar 4.5 berikut.



**Gambar 4. 5** Grafik Uji Coba Ke-2

Pada gambar 4.5 dapat dilihat bahwa pada uji coba ke-2 dengan menggunakan QO (*Query Optimization*) dapat mempercepat waktu akses (*Response time*) pada proses setiap langkah metode TOPSIS.

### 4.2.3 Uji Coba Ke-3

Uji coba ke-3 menggunakan data sebanyak 342 data yang sudah di inputkan pada database MariaDB. Pada tabel 4.5 terlihat bahwa uji coba ke-3 dilakukan 3 kali percobaan yang menghasilkan *response time* berbeda-beda pada setiap percobaan.

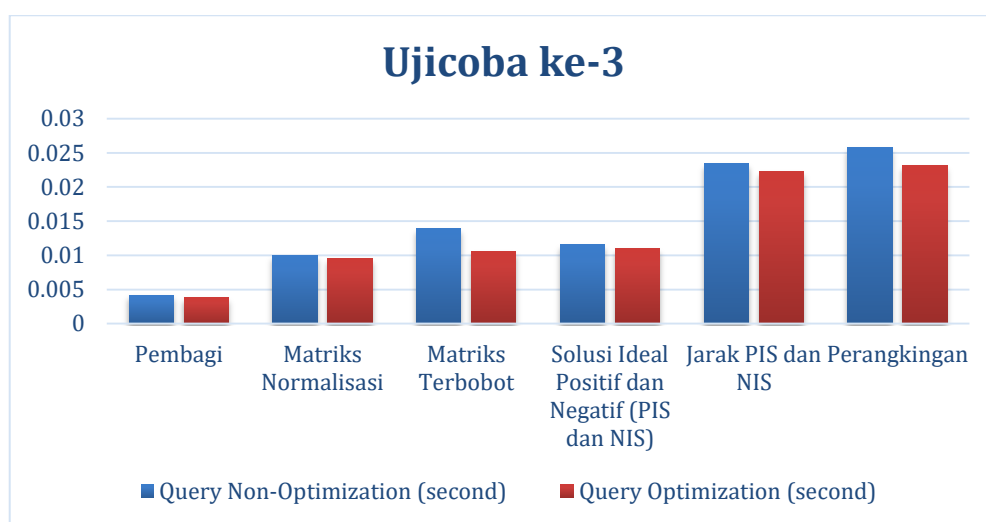
**Tabel 4. 5** Tabel Uji Coba Ke-3

Langkah TOPSIS	Percobaan						Average	
	1		2		3		QNO (second)	QO (second)
	QNO (second)	QO (second)	QNO (second)	QO (second)	QNO (second)	QO (second)		
1.	4	3.8	4.3	3.8	4	4.2	4.166	3.866
2.	9.3	8.9	10.1	9.9	10.5	9.8	9.966	9.533
3.	20.3	10.8	10.5	10.5	10.9	10.5	13.9	10.6
4.	10.8	10.8	11.4	11.3	12.5	11	11.566	11.033
5.	22.6	22	21.9	22.6	25.7	22.4	23.4	22.333
6.	31.2	24.2	23.2	22.5	23	22.7	25.8	23.133

QNO = *Query Non-Optimization*

QO = *Query Optimization*

Dari 3 kali percobaan pada uji coba ke-3 yang berjumlah 334 data diambil rata-rata (*average*) *response time* pada masing-masing langkah metode TOPSIS yang menggunakan QNO (*Query Non-Optimization*) dan QO (*Query Optimization*). Hasil dari *average* masing-masing langkah metode TOPSIS berdasarkan QNO (*Query Non-Optimization*) dan QO (*Query Optimization*) dapat dilihat pada gambar 4.7 berikut.



**Gambar 4. 6** Grafik Uji Coba Ke-3

Pada gambar 4.6 dapat dilihat bahwa pada uji coba ke-3 dengan menggunakan QO (*Query Optimization*) dapat mempercepat waktu akses (*Response time*) pada proses setiap langkah metode TOPSIS.

Berdasarkan hasil uji coba dengan menggunakan 3 data dengan banyaknya data yang berbeda, selain itu disetiap uji coba dilakukan 3 percobaan berdasarkan *query non-optimization* dan *query optimization*, maka ditemukan bahwa penerapan *query optimization* pada metode TOPSIS untuk sistem pendukung keputusan kerusakan sektor pasca bencana sangat bagus karena nilai rata-rata *response time query optimization* lebih kecil dibandingkan dengan nilai rata-rata

*response time query non-optimization*. Tabel 4.6 menunjukkan hasil nilai rata-rata *response time* dari hasil uji coba *query non-optimization* dan *query optimization*.

#### 4.2.4 Perbandingan Hasil Uji Coba

**Tabel 4. 6** Perbandingan Hasil Uji Coba

Langkah Metode TOPSIS	Uji coba ke-					
	1 (114 data)		2 (228 data)		3 (334 data)	
	QNO (Mili second)	QO (Mili second)	QNO (Mili second)	QO (Mili second)	QNO (Mili second)	QO (Mili second)
Pembagi	2.7	2.466	3.466	3.066	4.166	3.866
Matriks Normalisasi	4.133	3.133	6.2	6.066	9.966	9.533
Matriks Terbobot	5.2	4.033	8.333	7.3	13.9	10.6
Solusi Ideal Positif dan Negatif (PIS dan NIS)	7.4	6.333	9.1	8.1	11.566	11.033
Jarak PIS dan NIS	14.033	11.966	17.333	15.5	23.4	22.333
Perangkingan	11	10.8	16.566	15.8	25.8	23.133

Berdasarkan tabel diatas dapat ditarik suatu kesimpulan lagi yaitu semakin banyak data yang digunakan maka nilai *response time* yang dibutuhkan *query non-optimization* dan *query optimization* dalam memproses langkah-langkah metode TOPSIS semakin lama. Tetapi nilai rata-rata *response time query optimization* dalam memproses langkah-langkah TOPSIS untuk sistem pendukung keputusan kerusakan sektor bencana sangat bagus karena nilai rata-rata *response timenya* lebih kecil jika dibandingkan dengan rata-rata *response time query non-optimization*. Oleh karena itu perlu penerapan *query optimization* pada setiap pembuatan sistem atau aplikasi agar sistem atau aplikasi yang di bangun bisa lebih cepat *response timenya*.

Penerapan *query optimization* bisa dikembangkan pada penerapan metode artificial intelligence agar sistem atau aplikasi yang dibangun dapat lebih dinamis dan cepat respon timenya. Selain itu *query optimization* dapat diterapkan pada database NoSQL.

#### 4.2.5 Hasil Uji Coba Usability

Didapatkan sebanyak 30 responden yang memberikan penilaian berdasarkan lima point skala likert. Hasil kuesioner selanjutnya dihitung menggunakan rumus yang telah ditentukan untuk memperoleh Skor SUS. Hasil penilaian skor SUS ditampilkan pada Tabel 4.7.

**Tabel 4.7** Hasil Penilaian Skor SUS

Resp	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	SUS
1	3	0	3	1	4	1	4	1	4	0	52.5
2	4	1	3	1	4	1	3	1	3	2	57.5
3	3	1	3	1	3	1	3	1	3	1	50

4	4	0	3	0	4	0	4	0	3	0	45
5	3	1	3	1	3	1	3	1	3	1	50
6	3	1	3	1	3	1	3	1	3	1	50
7	3	1	3	1	3	1	3	1	3	0	47.5
8	3	1	3	1	3	1	3	1	3	1	50
9	4	1	2	1	2	1	3	1	4	1	50
10	3	1	3	1	3	1	3	1	3	2	52.5
11	3	1	3	2	2	1	3	2	2	2	50
12	3	1	3	1	3	1	3	1	3	2	52.5
13	3	1	3	1	3	1	3	1	3	2	52.5
14	3	1	3	1	3	1	3	1	4	0	50
15	3	1	4	1	3	1	4	0	3	1	52.5
16	4	0	4	0	4	0	4	0	2	1	47.5
17	3	2	2	2	2	1	3	2	3	1	52.5
18	3	1	3	1	3	1	3	1	3	1	50
19	2	1	3	2	3	2	2	1	3	1	47.5
20	3	1	3	1	3	1	3	1	3	1	50
21	3	1	3	1	3	1	3	1	3	1	50
22	3	1	3	1	3	1	3	1	3	1	50
23	3	1	3	1	3	1	3	1	3	1	50
24	3	1	3	0	4	1	3	2	3	1	52.5
25	3	1	3	1	3	1	3	1	3	1	50
26	3	1	3	1	3	1	2	2	3	1	50
27	3	0	3	1	3	1	3	1	3	1	47.5
28	3	1	2	1	3	2	3	0	3	1	47.5
29	3	0	3	1	3	0	4	1	3	1	47.5
30	3	1	3	1	4	0	4	0	3	1	50
<b>Rata-rata</b>											50.16667

Berdasarkan hasil pengujian dan perhitungan seluruh hasil kuisioner, didapatkan hasil yang menunjukkan rata-rata skor SUS sebesar 50.16667



Skor SUS dapat menunjukkan tingkat penerimaan pengguna. Skor SUS harus bernilai lebih dari 70 agar termasuk ke dalam kategori *Acceptable* (Brooke, 2013). Skor SUS DSS pada penelitian ini yaitu 50.16667.

### 4.3 Integrasi Islam

Penelitian ini bertujuan untuk menolong tim *suveyor* agar lebih mudah dan cepat dalam membantu korban bencana alam. Dengan adanya pertolongan ini, menunjukkan adanya hubungan antara manusia dengan manusia lain. Hal ini manusia diharuskan saling menolong antara sesamanya. Salah satu hadis yang dapat kita terapkan terkait tolong menolong yang diriwayatkan oleh Imam Muslim berikut ini:

مَنْ نَفَّسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرَابِ الدُّنْيَا نَفَّسَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرَابِ يَوْمِ الْقِيَامَةِ وَمَنْ يَسِّرْ عَلَى مُعْسِرٍ يَسِّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ وَمَنْ سَتَرَ مُسْلِمًا سَتَرَهُ اللَّهُ فِي الدُّنْيَا وَالْآخِرَةِ وَاللَّهُ فِي عَوْنِ الْعَبْدِ مَا كَانَ الْعَبْدُ فِي عَوْنِ أَخِيهِ.  
رَوَاهُ مُسْلِمٌ عَنْ أَبِي هُرَيْرَةَ

*"Barang siapa melapangkan seorang mukmin dari satu kesusahan dunia, Allah akan melapangkannya dari salah satu kesusahan di hari kiamat. Barang siapa meringankan penderitaan seseorang, Allah akan meringankan penderitaannya di dunia dan akhirat. Barang siapa menutupi (aib) seorang muslim, Allah akan menutupi (aib)nya di dunia dan akhirat. Allah akan menolong seorang hamba selama hamba itu mau menolong saudaranya."* (HR. Muslim).

Selain saling tolong menolong antara sesama manusia, penelitian ini akan mengingatkan manusia bagaimana manusia seharusnya menjaga hubungannya dengan Allah SWT. Manusia harus tetap mengingat tentang keberadaan Allah

SWT baik dalam suka cita maupun berduka, seperti yang dijelaskan pada QS Ar-Ra'd ayat 28 : yang berbunyi :

الَّذِينَ آمَنُوا وَتَطْمَئِنُّ قُلُوبُهُمْ بِذِكْرِ اللَّهِ أَلَا بِذِكْرِ اللَّهِ تَطْمَئِنُّ الْقُلُوبُ

*"(Yaitu) orang-orang yang beriman dan hati mereka manjadi tenteram dengan mengingat Allah. Ingatlah, hanya dengan mengingati Allah-lah hati menjadi tenteram."* (QS. Ar-Ra'd : 28)

Cara lain mengingat Allah juga bisa dengan mengamati dan mangagumi seluruh ciptaannya. Dengan ini, bisa juga dikatakan dengan mempererat hubungan manusia dengan alam yang juga ciptaan Allah SWT. Hal ini berkaitan juga dengan bencana alam yang menjadi tema penelitian ini. Mengingat untuk selalu memelihara dan bersyukur akan alam semesta yang telah Allah ciptakan dan bukan dengan merusaknya. Contohnya, jangan membuang sampah sembarangan ke sungai agar tidak sungai tidak meluap dan terjadi banjir. Selain itu, manusia juga harus menjaga hutan dengan tidak membakarnya hanya karena kepentingan pribadi. Hal ini menunjukkan adanya hubungan antara manusia dan alam sebagai ciptaan Allah SWT (Mu'amalah ma'a al-'alam). Seperti yang digambarkan pada QS Ar-Rum ayat 41 :

ظَهَرَ الْفَسَادُ فِي الْبَرِّ وَالْبَحْرِ بِمَا كَسَبَتْ أَيْدِي النَّاسِ لِيُذِيقَهُمْ بَعْضَ الَّذِي عَمِلُوا لَعَلَّهُمْ يَرْجِعُونَ

*"Telah tampak kerusakan di darat dan di laut disebabkan karena perbuatan tangan manusia; Allah menghendaki agar mereka merasakan sebagian dari (akibat) perbuatan mereka, agar mereka kembali (ke jalan yang benar)."* (QS. Ar-Rum : 41)

Menurut tafsir Jalalain (Harry, 2003) terkait QS Ar-Rum ayat 41, (Telah tampak kerusakan di darat) disebabkan terhentinya hujan dan menipisnya

tumbuh-tumbuhan (dan di laut) maksudnya di negeri-negeri yang banyak sungainya menjadi kering (disebabkan perbuatan tangan manusia) berupa perbuatan-perbuatan maksiat (supaya Allah merasakan kepada mereka) dapat dibaca *liyudziiqahum* dan *linudziiqahum*; kalau dibaca *linudziiqahum* artinya supaya Kami merasakan kepada mereka (sebagian dari akibat perbuatan mereka) sebagai hukumannya (agar mereka kembali) supaya mereka bertobat dari perbuatan-perbuatan maksiat.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil pengukuran *response time* yang sudah dilakukan, dapat disimpulkan bahwa dengan dilakukannya optimasi pada *query* TOPSIS didapatkan nilai *response time* yang lebih kecil dibandingkan nilai *response time* pada *query* TOPSIS non optimasi. Selain itu, didapati bahwa semakin banyak data yang digunakan maka nilai *response time* yang dibutuhkan untuk *query* non optimasi dan *query* optimasi dalam memproses langkah-langkah metode TOPSIS semakin lama. Oleh karena itu perlu diterapkan optimasi *query* pada setiap pengembangan sistem atau aplikasi agar sistem atau aplikasi yang kita bangun dapat memiliki *response time* yang lebih cepat.

Selanjutnya didapatkan skor dari pengujian *usability* sistem menggunakan SUS dengan 30 responden yaitu 50.16667 yang menunjukkan bahwa sistem masih belum *usable*. Hal ini menunjukkan bahwa sistem perlu dikembangkan lebih lanjut.

#### **5.2 Saran**

Penelitian ini dilakukan atas dasar pengembangan sistem yang telah dilakukan sebelumnya untuk mendapatkan tingkat akurasi yang lebih baik dengan metode yang lebih baik, namun peneliti menyadari bahwa penelitian ini masih jauh dari kata sempurna atau mendekati kebenaran, oleh karena itu adapun saran yang digunakan oleh peneliti adalah sebagai berikut:

1. Penelitian dapat dikembangkan lagi menggunakan Metode DSS lainnya sehingga dapat menghasilkan tingkat akurasi yang lebih tinggi dari penelitian

ini agar dapat menemukan metode yang cocok dalam penilaian tingkat kerusakan suatu sektor akibat bencana alam.

2. Sistem yang dibangun adalah berbasis website, oleh karena itu peneliti mengharapkan penelitian selanjutnya dapat digunakan berbasis mobile agar lebih *user friendly*.
3. Kekurangan dari penelitian ini yaitu masih belum dilengkapi dengan metode *artificial intelligence*, oleh karena itu disarankan untuk dapat dikembangkan lagi dengan mengimplementasikan metode *artificial intelligence* sehingga sistem akan lebih baik dalam memproses input.
4. Kekurangan lain pada penelitian ini yaitu tidak adanya pengukuran akurasi, oleh karena itu, disarankan untuk penelitian selanjutnya agar menambahkan uji coba untuk pengukuran akurasi sistem.

## DAFTAR PUSTAKA

- Almais, A. T. W., Fatchurrohman, ., Holle, K. F. H., Kinasih, K. S., Wiranti, D. A., & Yasin, S. Y. (2020). *Implementation Fuzzy Weighted Product Preparation Post Disaster Reconstruction and Rehabilitation Action based Dynamics Decision Support System*. 272–277.
- Bachriwindi, A., Putra, E. K., Munawaroh, U. M., & Almais, A. T. W. (2019). Implementation of Web-Based Weighted Product Use Decision Support System to Determine the Post-Disaster Damage and Loss. *Journal of Physics: Conference Series*, 1413(1).
- Brooke, J. (2013). *SUS : A Retrospective*. January 2013.
- Chakrabarti, K., Garofalakis, M., Rastogi, R., & Shim, K. (2001). Approximate query processing using wavelets. *VLDB Journal*, 10(2–3), 199–223.
- Eslami, M., Mahmoodian, V., Dayarian, I., Charkhgard, H., & Tu, Y. (2020). Query batching optimization in database systems. *Computers and Operations Research*, 121, 104983.
- Febriyati, M. N., Sophan, M. K., & Yunitarini, R. (2016). Recruitment Warga Laboratorium Teknik Informatika Di Universitas. *Jurnal SimanteC*, 5(3), 133–142.
- Gading Sadewo, M., Perdana Windarto, A., & Wanto, A. (2018). *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer) PENERAPAN ALGORITMA CLUSTERING DALAM MENGELOMPOKKAN BANYAKNYA DESA/KELURAHAN MENURUT UPAYA ANTISIPASI/ MITIGASI BENCANA ALAM MENURUT PROVINSI DENGAN K-MEANS*. 2, 311–319.
- H.N, I. A., Nugroho, P. I., & Ferdiana, R. (2015). Pengujian Usability Website Menggunakan System Usability Scale. *JURNAL IPTEKKOM : Jurnal Ilmu Pengetahuan & Teknologi Informasi*, 17(1), 31.
- Harry, M., As-suyuthi, J., Muhammad, J., & Ahmad, I. (2003). Al-Faatihah IR JALALAIN. *Mohammad Harty*.
- Ioannidis, Y. E. (2004). Query optimization. *Computer Science Handbook, Second Edition*, 28(1), 55-1-55–20.
- Kumari, N. (2012). SQL Server Query Optimization Techniques - Tips for Writing Efficient and Faster Queries. *International Journal of Scientific and Research Publications*, 2(6), 1–4.

- Kuo, T. (2017). A modified TOPSIS with a different ranking index. *European Journal of Operational Research*, 260(1), 152–160.
- Liang, W., Chen, B., & Yu, J. X. (2011). Top-k query evaluation in sensor networks under query response time constraint. *Information Sciences*, 181(4), 869–882.
- Mahdia, Faya; Noviyanto, F. (2013). PEMANFAATAN GOOGLE MAPS API UNTUK PEMBANGUNAN SISTEM INFORMASI MANAJEMEN BANTUAN LOGISTIK PASCA BENCANA ALAM BERBASIS MOBILE WEB ( Studi Kasus : Badan Penanggulangan Bencana Daerah Kota Yogyakarta ). *Jurnal Sarjana Teknik Informatika*, 1(1), 162–171.
- Martha, R., Firdaus, Y., & Laksitowening, K. A. (2010). Analisa Perbandingan Response Time Dan Throughput. *Seminar Nasional Aplikasi Teknologi Informasi 2010*, 2010(2010), E1–E5.
- Noviyanti, P., Deolika, A., Hartinah, S., Haris, C. A., Maryana, T., & Sari, N. D. (2018). Perbandingan Query Response Time pada Model Query View dan Cross Product. *E-Jurnal JUSITI (Jurnal Sistem Informasi Dan Teknologi Informasi)*, 7–2(2), 131–141.
- Purwanto, H., & Informatika, M. (2017). *SISTEM PENUNJANG KEPUTUSAN PEMILIHAN NOTEBOOK*. 2(2), 55–59.
- Schwarz, H., Wagner, R., & Mitschang, B. (2001). Improving the processing of decision support queries: The case for a DSS optimizer. *Proceedings of the International Database Engineering and Applications Symposium, IDEAS*, 177–186.
- Sharma, M., Singh, G., & Singh, R. (2018). Clinical decision support system query optimizer using hybrid Firefly and controlled Genetic Algorithm. *Journal of King Saud University - Computer and Information Sciences*.
- Sharma, Manik, Singh, G., & Singh, R. (2016). Design and analysis of stochastic DSS query optimizers in a distributed database system. *Egyptian Informatics Journal*, 17(2), 161–173.
- Wibowo Almais, A. T., Sarosa, M., & Muslim, M. A. (2016). Implementation Of Multi Experts Multi Criteria Decision Making For Rehabilitation And Reconstruction Action After A Disaster. *Matics*, 8(1), 27.
- Yana, M. S., Setiawan, L., Ulfa, E. M., & Rusyana, A. (2018). Penerapan Metode K-Means dalam Pengelompokan Wilayah Menurut Intensitas Kejadian Bencana Alam di Indonesia Tahun 2013-2018. *Journal of Data Analysis*, 1(2), 93–102.
- Yu, D., & Pan, T. (2021). Tracing knowledge diffusion of TOPSIS: A historical

perspective from citation network. *Expert Systems with Applications*, 168(November), 114238.