

7019-Article Text-48860-1-6-
20250907____.docx

by 1 1

Submission date: 12-Sep-2025 02:36PM (UTC+0530)

Submission ID: 2748761931

File name: 7019-Article_Text-48860-1-6-20250907____.docx (742.1K)

Word count: 4335

Character count: 28767

Perancangan *Stored Procedure* MySQL untuk Implementasi Algoritma Simple Additive Weighting

Tri Hastono^{1*}, Prahenusa Wahyu Ciptadi², Crismantoro Budisaputro³, Nayaka Anung Fahriza⁴

^{1,2}Informatika, Universitas PGRI Yogyakarta, Bantul, Indonesia

³Rekam Medis, STIKES Bhakti Husada Mulia Madiun, Madiun, Indonesia

⁴TKIT Al Farabi, TKIT Al Farabi, Bantul, Indonesia

Email: ^{1*}trihastono@gmail.com, ²nusa@upy.ac.id, ³crismantoro@stikes-bhm.ac.id, ⁴nayakaanungf@gmail.com

Email Penulis Korespondensi: trihastono.13@gmail.com

Abstrak– Dosen memiliki peran strategis dalam pendidikan tinggi, namun beban kerja yang kompleks menimbulkan tantangan sehingga keberadaan asisten dosen menjadi solusi penting. Seleksi asisten dosen yang masih dilakukan secara manual menghadapi kendala berupa keterbatasan waktu, subjektivitas, dan ketidakakuratan. Penelitian terdahulu telah memanfaatkan metode *Simple Additive Weighting (SAW)* untuk mendukung seleksi multikriteria, namun umumnya diterapkan pada tingkat aplikasi sehingga rawan inkonsistensi, redundansi kode, serta masalah keamanan. Penelitian ini bertujuan merancang *Stored Procedure* basisdata untuk mengotomatisasi algoritma SAW pada proses seleksi asisten dosen langsung di level basis data. Metode yang digunakan adalah model *Waterfall* melalui tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian menggunakan data simulasi. *Store Procedure* yang dikembangkan mencakup fungsi normalisasi, pembobotan, dan pemeringkatan sehingga proses seleksi lebih efisien, konsisten, dan terstandarisasi. Hasil pengujian menunjukkan bahwa perhitungan otomatis menghasilkan nilai preferensi identik dengan perhitungan manual, dengan kandidat “Nayaka Anung” memperoleh skor 30,93 dan peringkat ketiga. Temuan ini membuktikan bahwa penelitian telah sesuai dengan tujuan yang dirumuskan serta memberikan kontribusi nyata dalam menghadirkan mekanisme seleksi yang lebih objektif, akurat, dan dapat diandalkan bagi institusi pendidikan tinggi.

Kata Kunci: Desain, Store Procedure, Otomatisasi, Asisten Dosen, Simple Additive Weighting (SAW).

Abstract– Lecturers hold a strategic role in higher education, yet their complex workload often creates challenges, making the presence of teaching assistants an essential solution. The selection process for teaching assistants, which is still carried out manually, faces obstacles such as time constraints, subjectivity, and inaccuracy. Previous studies have utilized the Simple Additive Weighting (SAW) method to support multi-criteria selection, but most were applied only at the application level, leading to potential inconsistencies, code redundancy, and security issues. This study aims to design a MySQL Stored Procedure to automate the SAW algorithm for teaching assistant selection directly at the database level. The development method employed was the Waterfall model, consisting of requirements analysis, design, implementation, and testing with simulated data. The developed Stored Procedure includes functions for normalization, weighting, and ranking, thereby ensuring a more efficient, consistent, and standardized process. Testing results indicated that automated calculations produced preference values identical to manual calculations, with the candidate “Nayaka Anung” obtaining a score of 30.93 and ranking third. These findings confirm that the study has achieved its intended objectives and contributes significantly by providing a more objective, accurate, and reliable mechanism for teaching assistant selection in higher education institutions.

Keywords: Design, Store Procedure, Automation, Teaching Assistant, Simple Additive Weighting (SAW).

1. PENDAHULUAN

Dosen adalah sebuah Profesi yang tidak asing di telinga masyarakat, terlebih lagi di dunia pendidikan tinggi. Dosen merupakan salah satu profesi yang memiliki peran strategis dalam membentuk generasi penerus bangsa melalui pendidikan, penelitian, dan pengabdian kepada masyarakat [1], [2]. Di era digital seperti saat ini, peran dosen semakin kompleks karena tidak hanya dituntut untuk menguasai materi ajar, tetapi juga mampu menyesuaikan metode pengajaran dengan perkembangan teknologi dan kebutuhan industri. Sebagai figur akademik, dosen juga diharapkan menjadi teladan dan sumber inspirasi bagi mahasiswa dalam mengembangkan kompetensi, kreativitas, serta etika profesional. Oleh karena itu, profesi dosen bukan sekadar pekerjaan mengajar, melainkan sebuah pengabdian yang menuntut dedikasi, kompetensi, dan kemampuan beradaptasi terhadap berbagai tantangan [2]–[4].

Tugas dosen tidak terbatas pada kegiatan mengajar di kelas saja. Tri Dharma Perguruan Tinggi, yang mencakup pendidikan, penelitian, dan pengabdian kepada masyarakat, menjadi tanggung jawab utama yang harus dilaksanakan secara seimbang dan berkualitas [3]–[6]. Di samping itu, beban administratif yang semakin kompleks, seperti pelaporan kegiatan akademik, persiapan akreditasi, serta penyusunan laporan penelitian dan pengabdian, kerap menguras waktu dan energi. Tidak jarang, dosen juga dihadapkan pada tuntutan untuk berinovasi dalam pengajaran, mengembangkan bahan ajar berbasis teknologi, dan berpartisipasi aktif dalam berbagai forum akademik. Ditambah dengan peran dan tanggung jawab di ranah keluarga, beban kerja yang menumpuk ini menjadikan manajemen waktu sebagai tantangan besar bagi para dosen [4], [6], [7].

Di sisi lain, kesejahteraan dosen sering kali tidak sebanding dengan beratnya beban pekerjaan yang harus dipikul. Remunerasi yang diterima tidak selalu menggambarkan terhadap dedikasi dan kerja keras yang dilakukan. Dan itu terlihat sekali bagi dosen yang masih berada pada jenjang awal karier [8]. Kondisi ini mendorong banyak dosen untuk mencari pekerjaan tambahan di luar kampus guna menambah penghasilan. Namun, situasi tersebut sering kali berdampak pada

berkurangnya fokus dan waktu yang seharusnya dialokasikan untuk menjalankan tugas akademik, seperti membimbing mahasiswa, mengembangkan penelitian, atau berkontribusi dalam pengabdian kepada masyarakat. Ketidakeimbangan ini pada akhirnya dapat mengurangi kualitas pelayanan akademik yang diberikan kepada mahasiswa dan *stakeholders* kampus[6], [8], [9].

Selain faktor kesejahteraan, dinamika politik di lingkungan kerja juga memberikan pengaruh signifikan terhadap beban kerja dosen. Fenomena ini dapat terlihat dari adanya ketidakadilan dalam pembagian beban mengajar, penugasan administrasi yang tidak merata, hingga pengaruh senioritas yang kadang menghambat kesempatan pengembangan karier dosen junior. Dalam beberapa kasus, politik internal ini juga dapat memengaruhi penentuan akses terhadap fasilitas penelitian atau proyek-proyek strategis di kampus[10], [11]. Ketidakstabilan situasi ini berpotensi menurunkan motivasi dosen dan menghambat tercapainya kinerja optimal yang mendukung visi dan misi institusi pendidikan tinggi[8], [9].

Untuk menjawab berbagai tantangan tersebut, keberadaan asisten dosen menjadi salah satu solusi yang sangat membantu. Asisten dosen, yang biasanya direkrut dari mahasiswa berprestasi di program studi terkait, dapat membantu meringankan beban kerja dosen dalam banyak aspek. Tugas asisten dosen dapat mencakup membantu menyiapkan materi ajar, mendampingi praktikum, mengoreksi tugas[8] hingga mengelola administrasi perkuliahan. Kehadiran mereka tidak hanya mengurangi tekanan kerja dosen, tetapi juga memberikan kesempatan kepada mahasiswa untuk mengembangkan keterampilan mengajar, memperluas pemahaman materi, dan mendapatkan pengalaman praktis di lingkungan akademik. Dengan sinergi yang baik antara dosen dan asisten, kualitas layanan pendidikan dapat meningkat secara signifikan[12], [13].

Namun, penentuan asisten dosen tidak dapat dilakukan secara sembarangan. Proses seleksi harus bersifat adil, transparan, dan berbasis kriteria yang jelas agar menghasilkan keputusan yang objektif. Beberapa kriteria yang biasanya digunakan antara lain prestasi akademik, keterampilan mengajar, kemampuan teknis, kepribadian dan *soft skill*, pengalaman, serta tingkat komitmen terhadap tanggung jawab yang akan diemban. Ketepatan dalam pemilihan asisten dosen sangat krusial karena akan mempengaruhi kelancaran pelaksanaan kegiatan akademik dan kepuasan kedua belah pihak, baik dosen sebagai pembimbing maupun mahasiswa yang terlibat sebagai asisten[12], [13].

Dalam praktiknya, proses seleksi asisten dosen yang dilakukan secara manual sering kali menghadapi berbagai kendala, seperti keterbatasan waktu, potensi subjektivitas, serta ketidakakuratan dalam penilaian kandidat [14]–[16]. Sebagai contoh mengenai keterbatasan waktu, di mana pengelola harus menyeleksi banyak kandidat dalam periode yang relatif singkat. Kondisi ini dapat menyebabkan proses seleksi berjalan tidak efisien dan berpotensi mengurangi ketelitian dalam mengevaluasi setiap kriteria yang ditetapkan. Selain itu, proses manual juga memperbesar risiko terjadinya inkonsistensi karena tidak adanya standar baku yang seragam dalam menilai kandidat. Faktor subjektivitas juga menjadi tantangan tersendiri yang tidak dapat dihindari dalam seleksi manual. Penilaian yang didasarkan pada persepsi individu berpotensi menimbulkan ketidakadilan, sehingga hasil seleksi kurang mencerminkan kualitas kandidat yang sebenarnya. Hal ini diperparah dengan kemungkinan terjadinya ketidakakuratan dalam proses penilaian, baik karena kesalahan pencatatan, kurangnya data pendukung, maupun keterbatasan instrumen evaluasi. Kendala-kendala tersebut membuat seleksi manual rawan menghasilkan keputusan yang kurang optimal, sehingga dibutuhkan pendekatan baru yang lebih sistematis dan objektif [14]–[17], [18].

Terdapat beberapa penelitian terdahulu yang menjadi rujukan pada penelitian ini. Penelitian-penelitian tersebut telah menerapkan algoritma *Simple Additive Weighting* (SAW) dengan model komputerisasi untuk mendukung proses seleksi asisten dosen ataupun kasus multikriteria lainnya [17], [19]–[23]. Salah satu contoh adalah penelitian yang diangkat oleh Triana Elizabeth dan Tinaliah [17], penelitian tersebut dilatarbelakangi masalah minimalnya dampak kurang positif dari asisten dosen yang terpilih dan proses pemilihan berdasarkan urutan nomor pendaftaran. Solusi yang ditawarkan dari penelitian Triana Elizabeth dan Tinaliah adalah dengan menghadirkan aplikasi untuk seleksi asisten dosen berbasis komputer menggunakan *Fuzzy Simple Additive Weighting* (SAW). Perancangan aplikasi pada penelitian tersebut menggunakan aplikasi Netbeans IDE 8.2 dan MySQL.

Meskipun demikian, penelitian-penelitian terdahulu masih terbatas pada implementasi algoritma *Simple Additive Weighting* (SAW) di tingkat aplikasi (*application layer*). Pendekatan ini memiliki sejumlah kelemahan, di antaranya redundansi kode, keterbatasan efisiensi, risiko inkonsistensi data, serta kerentanan terhadap serangan *SQL Injection* [24], [25]. Sejauh yang peneliti ketahui, dalam konteks pemilihan asisten dosen, belum ada penelitian yang mengangkat secara khusus pengintegrasian algoritma *Simple Additive Weighting* (SAW) langsung di tingkat basis data melalui *Stored procedure*.

Penelitian ini berusaha menjawab permasalahan yang ada melalui *Store Procedure* basisdata. Pendekatan ini memungkinkan proses perhitungan dilakukan langsung pada level basis data, sehingga meningkatkan efisiensi, keamanan, dan konsistensi hasil seleksi. *Store Procedure* yang dirancang pada penelitian ini adalah *function* dan *procedure*. Pada penelitian ini terdapat 2 kontribusi baru dari penelitian pendahulu. Pertama, kontribusi baru penelitian ini terletak pada lokasi logika algoritma *Simple Additive Weighting* (SAW). Pada penelitian ini logika algoritma *Simple Additive Weighting* (SAW) berada di *Store Procedure* basisdata, sehingga aplikasi hanya berfungsi sebagai media input data tanpa perlu menangani proses perhitungan. Untuk kontribusi baru yang lain adalah *Store Procedure* yang dirancang dengan lengkap (normalisasi, pembobotan, penentuan peringkat akhir menggunakan SAW), sehingga *Store Procedure* bisa langsung dijalankan secara otomatis menggunakan sintaks khusus. Pendekatan ini tidak hanya meningkatkan efisiensi dan konsistensi proses seleksi asisten dosen, tetapi juga memperkuat aspek keamanan dan keandalan sistem.

2. METODOLOGI PENELITIAN

7

8

Selain wawancara dengan Ketua Program Studi Informatika, penelitian ini juga menggunakan bahan pendukung lain untuk memperkuat kebutuhan sistem. Bahan pendukung tersebut meliputi dokumen akademik kampus seperti pedoman akademik, aturan seleksi asisten dosen, serta data historis asisten dosen pada periode sebelumnya. Dokumen ini digunakan untuk memvalidasi kriteria dan bobot yang diperoleh melalui wawancara, sehingga hasil yang didapatkan tidak hanya bersifat subjektif tetapi juga sesuai dengan regulasi institusi. Untuk memperkaya data penelitian, peneliti juga melakukan studi literatur dari penelitian terdahulu terkait seleksi asisten dosen berbasis algoritma *Simple Additive Weighting* (SAW) maupun metode pengambilan keputusan multicriteria. Studi literatur ini digunakan untuk mengidentifikasi kriteria umum yang relevan dalam pemilihan asisten dosen, kemudian disesuaikan dengan konteks lokal Program Studi Informatika.

Tahap kedua adalah perancangan system. Pada tahap ini menghasilkan model logika algoritma *Simple Additive Weighting* (SAW) dalam bentuk *flowchart*, desain struktur tabel, dan rancangan query SQL yang akan dikompilasi ke dalam *Stored Procedure*. Dalam tahap ini, dilakukan pemetaan setiap langkah algoritma *Simple Additive Weighting* (SAW), seperti normalisasi nilai, pembobotan, dan penjumlahan tertimbang ke dalam bahasa SQL prosedural. Rancangan *Stored Procedure* disusun agar bersifat modular, efisien, dan aman, sehingga meminimalkan risiko kesalahan dan memudahkan pengujian ulang.

Tahap terakhir adalah implementasi dan pengujian, di mana *Stored Procedure* yang telah dirancang diimplementasikan ke dalam server MySQL. Prosedur ini kemudian diuji menggunakan data uji berupa alternatif dan kriteria dengan variasi 20 ran, untuk mengukur waktu eksekusi dan keakuratan hasil peringkat. Pengujian dilakukan dengan membandingkan hasil perhitungan *Simple Additive Weighting (SAW)* secara manual dengan hasil yang dihasilkan oleh *Stored Procedure*. Hasil dari tahap ini menjadi dasar untuk mengevaluasi keberhasilan strategi otomatisasi yang diusulkan dalam penelitian ini. Gambar 1 dibawah ini adalah gambar alur penelitian ini.



Gambar 1. Alur Penelitian

2.2 Data Penelitian

Hasil dari proses wawancara dengan ketua Program Studi Informatika didapatkan beberapa kebutuhan system. Kebutuhan system yang dimaksud adalah data alternatif, kriteria beserta bobotnya, data program studi, data fakultas dan data. Untuk data alternatif adalah daftar mahasiswa yang masuk untuk kriteria sebagai asisten dosen. Dan untuk data kriteria disajikan pada tabel 1.

Tabel 1. Data Kriteria Penelitian

No	Kode	Nama Kriteria	Indikator	Bobot
1	CRT1	IPk	Benefit	2.6
2	CRT2	Keterampilan Komunikasi	Benefit	4.0
3	CRT3	Keterampilan Menyampaikan Materi Secara Efektif	Benefit	4.1
4	CRT4	Kemampuan Berkolaborasi	Benefit	4.5
5	CRT5	Pengetahuan Teknologi	Benefit	2.9
6	CRT6	Kemampuan Bekerja dalam Tim	Benefit	3.8
7	CRT7	Kemampuan Menilai Pekerjaan Mahasiswa	Benefit	3.3
8	CRT8	Keterampilan Memberikan Bimbingan Akademis	Benefit	4.7
9	CRT9	Keterampilan Memberikan Bimbingan Akademis	Benefit	4.3
10	CRT10	Pengalaman dalam kegiatan penelitian atau proyek yang relevan	Benefit	3.6
11	CRT11	Kemampuan Menangani Tantangan	Benefit	3.4
12	CRT12	Keaktifan dalam Organisasi	Cost	3.0
13	CRT13	Jarak Domisili dengan Kampus	Cost	2.8

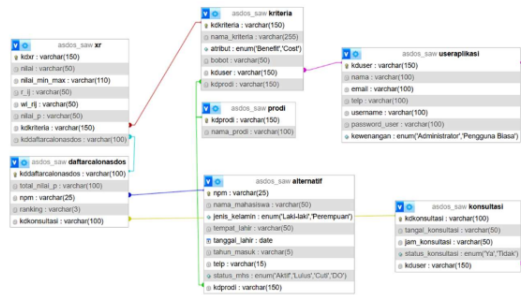
Berdasarkan hasil wawancara yang dilakukan, isi dari bobot kriteria asisten dosen mengacu pada nilai rentang 0-5. Narasumber menyampaikan *Store Procedure* yang dibuat dapat diadopsi pada sebuah aplikasi nyata.

21
3. HASIL DAN PEMBAHASAN

Seperti yang sudah disampaikan pada bagian awal mengenai fokus penelitian, dimana fokus penelitian pada perancangan *Store Procedure* untuk otomatisasi proses hitung algoritma Simple Additive Weighting. *Store Procedure* yang pada penelitian ini dibangun pada DMBS MySQL dan jenis *Store Procedures* yang dipilih pada penelitian adalah *function* dan *procedure*.

3.1 Struktur basisdata penelitian

Basisdata memainkan peran sentral sebagai tempat penyimpanan dan pengelolaan data yang diperlukan oleh aplikasi system informasi. Integrasi dengan basisdata memungkinkan sistem untuk mengakses informasi secara efisien, menyajikan alternatif keputusan berdasarkan data yang akurat, dan mendukung analisis yang dalam. Untuk desain tabel dalam basis basisdata pada penelitian ini disajikan pada gambar 2 dibawah ini



32
Gambar 2. Basisdata Penelitian

Basis data yang dirancang dalam penelitian ini terdiri dari tujuh tabel utama yang saling terintegrasi untuk mendukung proses implementasi algoritma Simple Additive Weighting (SAW) dalam penentuan asisten dosen. Untuk deskripsi 7 tabel utama tersebut disajikan pada tabel 2.

Tabel 2. Deskripsi tabel penelitian

No	Nama Tabel	Deskripsi Tabel
1	prodi	Tabel prodi merupakan tabel dalam penelitian yang difungsikan untuk menyimpan data program studi
2	kriteria	Tabel ini digunakan untuk menyimpan data kriteria calon asisten dosen beserta bobotnya
3	alternatif	Tabel alternatif digunakan untuk menyimpan data mahasiswa yang berpotensi sebagai kadindat asisten dosen
4	useraplikasi	Tabel useraplikasi ini digunakan untuk menyimpan pengguna aplikasi baik administrator atau yang lainnya yang memiliki kewenangan menggunakan aplikasi.
5	konsultasi	Tabel konsultasi digunakan untuk menyimpan rekaman perhitungan yang dilakukan oleh pengguna aplikasi
6	xr	Tabel xr ini difungsikan untuk menyimpan data matrik nilai calon asisten dosen terhadap kriteria beserta bobotnya
7	daftarcalonasdos	Tabel xr ini difungsikan untuk menyimpan data matrik nilai calon asisten dosen terhadap kriteria beserta bobotnya

tabel-
basisdata
berhasil
maka

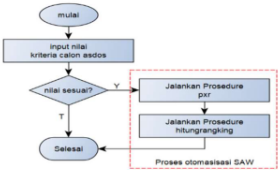
selanjutnya adalah melakukan pembuatan *Stored Procedure* untuk otomatisasi algoritma *Simple Additive Weighting* (SAW). (SAW).

3.2 Store Procedure Penelitian

Stored Procedure adalah sekumpulan perintah SQL yang disimpan dan dijalankan di dalam server basis data secara terstruktur dan otomatis. Prosedur ini dirancang untuk menjalankan tugas-tugas tertentu, seperti perhitungan, manipulasi

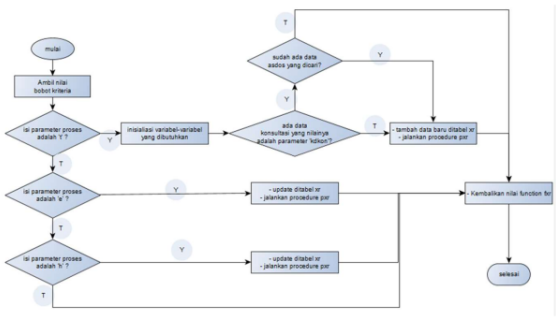
data, atau proses pengolahan informasi, tanpa perlu menulis ulang kode SQL setiap kali proses dijalankan. Dalam penelitian ini, *Stored Procedure* dirancang untuk mengotomatisasi proses seleksi asisten dosen dengan mengakses tujuh tabel yang telah terstruktur dalam basis data MySQL. Hal ini berarti kerja algoritma *Simple Additive Weighting* (SAW) tidak lagi berada pada aplikasi tapi berada pada level basisdata.

Proses otomatisasi algoritma *Simple Additive Weighting* (SAW) untuk menentukan asisten dosen diawali dari proses input nilai kriteria dari setiap calon asisten dosen. Proses input nilai ini dirancang agar dilakukan penghitungan matrik ternormalisasi yang disesuaikan dengan jenis dari kriterianya (*benefit* atau *cost*), perhitungan preferensi, dan pengurutan peringkat berdasarkan nilai akhir preferensi. Karena ada 3 proses utama, maka pada penelitian ini dalam membuat *Stored Procedure* dilakukan secara modularitas. Secara umum, *Store Procedure* yang digunakan untuk penentuan asisten dosen secara otomatis yang merupakan implementasi algoritma *Simple Additive Weighting* (SAW) digambarkan pada gambar 3.



Gambar 3. Flowchart implementasi algoritma SAW

Untuk Proses input nilai kriteria calon asdos sendiri menggunakan *store procedures* berjenis *function*. Alasan utamanya adalah *function* adalah salah satu jenis *Store Procedures* yang memberikan nilai kembalian. Nilai kembalian ini digunakan untuk informasi status keberhasilan maupun kegagalan proses ketika dilakukan proses input nilai kriteria calon asisten dosen. *Flowchart* dari *function* untuk input nilai kriteria calon asisten dosen dapat dilihat pada gambar 4.



Gambar 4. Flowchart dari function untuk input nilai kriteria calon asisten dosen

Dari flowchart yang ditunjukkan pada gambar 4 diatas, kemudian dituangkan pada sintaks *Store Procedure* MySQL. *Function* *fxr* adalah nama *function* yang merupakan implementasi dari proses input nilai kriteria asisten dosen. Untuk sintaks dari *function* *fxr* dapat dilihat pada gambar 5 dibawah.

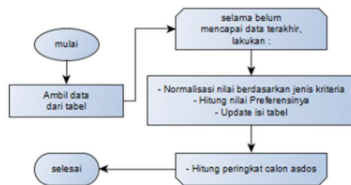
```

410 create function fxr(proses varchar(1),kdkon varchar(100),nil varchar(100),nmkrit varchar(100),kdalt varchar(100),prdt varchar(100),kd varchar(200)) returns varchar(10)
411 begin
412 set @Robotatritribot=(SELECT distinct(k.kobot) from kriteria k, prdt p where k.kgprodi=p.kgprodi and k.nama_kriteria=nmkrit and p.nama_prodi=prdt);
413 if proses='t' then
414 begin
415 set @Robotatritribot=(SELECT count(*) from sr s,@defaralnomasid d,konsultasi k where
416 s.kiddefaralnomasid=d.kiddefaralnomasid and d.kidkonsultasi=k.kidkonsultasi and k.kidnmkrit=@kdalt);
417 set @Robotatritribot=(SELECT count(*) from sr s,@defaralnomasid d,konsultasi k where s.sj=(SELECT distinct(kid) from k where k.kidprodi=p.kidprodi and k.kidkriteria=nmkrit and p.nama_prodi=prdt);
418 set @Robotatritribot=(SELECT distinct(k.kobot) from kriteria k, prdt p where k.kgprodi=p.kgprodi and k.kidkriteria=nmkrit and p.nama_prodi=prdt);
419 set @Robotatritribot=(SELECT distinct(k.kobot) from defaralnomasid d where s.kidalt and k.kidkonsultasi=k.kidkonsultasi);
420 if @Robotatritribot=0 then --jika TIDAK ADA RECORD ID KONSULTASI --
421 begin
422 insert into sr values(@Robotatritribot,nil,nil,nil,@Robotatritribot,nil,@kdalt,@kd);
423 call per(kdkon);
424 end;
425 else --jika ADA RECORD ID KONSULTASI --
426 begin
427 set @Robotatritribot=(SELECT count(*) from sr s,kriteria k,prdt p,@defaralnomasid d,konsultasi k
428 where s.kiddefaralnomasid=d.kiddefaralnomasid and d.kidkonsultasi=k.kidkonsultasi and k.kidnmkrit=@kdalt and k.kidkriteria=nmkrit and p.nama_prodi=prdt);
429 if @Robotatritribot=0 then
430 begin
431 insert into sr values(@Robotatritribot,nil,nil,nil,@Robotatritribot,nil,@kdalt,@kd);
432 call per(kdkon);
433 end;
434 else if;
435 end;
436 return concat(proses,cou_count());
437 end;
438 elseif proses='e' then
439 begin
440 update sr set nila=nil where kdkon=@kd;
441 call per(kdkon);
442 return concat(proses,cou_count());
443 end;
444 elseif proses='h' then
445 begin
446 delete from sr where kdkon=@kd;
447 call per(kdkon);
448 return concat(proses,cou_count());
449 end;
450 end if;
451 end if;
452 return @kd;

```

Gambar 5. Sintaks lengkap function fxr

Dari gambar 5 diatas, bisa diketahui bahwa terdapat 7 parameter input dari *function* fxr, 7 parameter input tersebut adalah proses, kdkon, nil, nmkrit, kdalt, prdt, dan parameter kd. Isi dari parameter-paramter input ini digunakan pada proses selanjutnya. Parameter proses pada *function* fxr hanya menerima hanya akan bekerja ketika berisi huruf 't' (**tambah data**), 'e' (**edit data**), dan 'h' (**hapus data**). Parameter kdkon digunakan untuk menyimpan nilai kode/id konsultasi. Ketika pengguna aplikasi melakukan akan melakukan penentuan asisten dosen, pengguna aplikasi tersebut akan tercatat melakukan konsultasi. Dan konsultasi yang dilakukan disimpan tabel konsultasi dan diberi kode unik dalam tabel tersebut. Setelah itu kode unik tersebut digunakan untuk proses penentuan asisten dosen. Parameter nil digunakan untuk menyimpan data nilai kriteria, dari asisten dosen yang dimasukan oleh pengguna aplikasi. Parameter nmkrit adalah sebuah parameter dari *function* fxr yang digunakan untuk menyimpan data kriteria yang dipilih oleh pengguna aplikasi. Parameter kdalt digunakan untuk menyimpan data calon asisten dosen. Parameter prdt digunakan untuk menyimpan data program studi. Dan parameter kd digunakan untuk menyimpan data kode unik dari tabel xr. Dari nilai setiap parameter input *function* fxr tersebut digunakan untuk proses selanjutnya. *Function* fxr ini akan menjalankan proses-proses didalamnya jika parameter utama proses adalah 't' atau 'e' atau 'h'. Huruf-huruf ini untuk sengaja dibuat untuk mewakili proses *Create Read Update Delete* (CRUD) dalam tabel basisdata. Huruf 't' yang berarti proses tambah data, huruf 'e' yang berarti proses edit data, sedangkan untuk huruf 'h' adalah untuk proses hapus data. Ada satu sintaks yang sama apapun isi dari parameter input proses, yakni sintaks dibaris 410, 421, 431, dan 437. Baris-baris ini adalah sintaks untuk menjalankan *store procedure* berjenis *procedure* dengan nama *procedure* pxx. *Procedure* pxx adalah *procedure* yang dibuat untuk proses utama, yaitu penentuan asisten dosen menggunakan *Simple Additive Weighting* (SAW). (SAW). Karena *procedure* pxx inilah yang merupakan *store procedure* utama, maka proses-proses algoritma *Simple Additive Weighting* (SAW). (SAW) dituliskan pada *procedure* ini. Proses tersebut adalah normalisasi input dan penghitungan preferensinya. *Flowchart* pada *procedure* pxx digambarkan pada gambar 6.



Gambar 6. Flowchart Procedure pxx

Dari gambar 6 diatas, terlihat *procedure* ini direncanakan melakukan perulangan proses. Proses-proses tersebut adalah proses algoritma *Simple Additive Weighting* (SAW). (SAW). Gambar 7 dibawah adalah sintaks keseluruhan *procedure* pxx.

```

356 drop procedure if exists ppx;
357 DELIMITER //
358 create procedure ppx(kriteria varchar(255))
359 begin
360     DECLARE selesai INT UNSIGNED DEFAULT 0;
361     DECLARE kode_xr varchar(15); DECLARE **;
362     DECLARE nilai varchar(15) DEFAULT **;
363     DECLARE nilai_max varchar(15) DEFAULT **;
364     DECLARE waji varchar(15) DEFAULT **;
365     DECLARE atb varchar(15) DEFAULT **;
366     DECLARE ranking varchar(15) DEFAULT 0;
367
368     DECLARE d_kursor CURSOR FOR SELECT distinct(x.kode),x.nilai,x.waji,x.atb,
369     (select max(nilai) from xr where kdkriteria=x.kdkriteria)
370     from xr x,kriteria k,daftarcalonasdos d where x.kdkriteria=k.kdkriteria and
371     x.kodecalonasdos=d.kodecalonasdos and
372     d.kidnasmen=kidnasmen;
373
374     DECLARE CONTINUE HANDLER FOR NOT FOUND SET selesai = 1;
375
376     OPEN d_kursor;
377     model_xmr LOOP
378     fetch d_kursor INTO kode_xr,nilai,waji,atb,nilai_max;
379     if selesai = 1 then
380         leave model_xmr;
381     end if;
382     update xr set nilai_max=nilai_max,
383     "1"||if(atb="benefit","(nilai/nilai_max),(nilai_max/nilai)",
384     nilai_max*((if(atb="benefit","(nilai/nilai_max),(nilai_max/nilai))"*(waji)) where kode=kode_xr;
385     END LOOP model_xmr;
386     CLOSE d_kursor;
387     call hitungranking(kriteria);
388     update kidnasmen set status_kidnasmen="Pa" where kidnasmen=kidnasmen;
389 end if;
390 DELIMITER ;

```

Gambar 7. Sintaks lengkap *procedure ppx*

Dari gambar 7 diatas, baris yang menunjukkan perulangan proses berada pada baris 378-386. Untuk proses normalisasi nilai kriteria setiap asisten dosen yang disesuaikan dengan jenis dari kriteria tersebut terletak pada baris 383-385. Pada baris 383-385 terlihat jelas dilakukan update data pada tabel xr dan variable *r_ij* diisi dengan hasil perhitungan jika isi dari variable atb “benefit” ataupun “cost”. Masih pada baris tersebut dilakukan proses perhitungan nilai preferensi setiap asisten dosen (baris 385) per kriteria. Nilai preferensi lah yang menentukan peringkat masing-masing calon asisten dosen. Dibagian akhir dari *procedure ppx* melakukan proses pemanggilan *procedure* hitungranking.

Procedure hitungranking dirancang untuk melakukan update data dari tabel daftarcalonasdos. Di tabel daftarcalonasdos tersimpan asist³ dosen yang sudah diurukan berdasarkan total nilai preferensi. Untuk *flowchart* dari *procedure* hitungranking dapat dilihat pada gambar 8 dibawah



Gambar 8. *Flowchart* dari *procedure* hitungranking

dari gambar 9 diatas, *procedure* hitungranking juga dilakukan perulangan proses. Proses tersebut adalah update isi tabel daftarcalonasdos. Untuk gambar 9 dibawah ini adalah sintaks lengkap *procedure* hitungranking.

```

391 drop procedure if exists hitungranking;
392 DELIMITER //
393 create procedure hitungranking(kriteria varchar(255))
394 begin
395     DECLARE selesai INT UNSIGNED DEFAULT 0;
396     DECLARE np varchar(15) DEFAULT **;
397     DECLARE ranking varchar(15) DEFAULT 0;
398     DECLARE totalp varchar(15) DEFAULT **;
399
400     DECLARE d_kursor CURSOR FOR
401     SELECT distinct(x.kode),max(nilai),jika total_p
402     from xr x,daftarcalonasdos d
403     where
404     x.kodecalonasdos=d.kodecalonasdos and
405     d.kidnasmen=kriteria.kidnasmen order by d.np order by total_p desc;
406     DECLARE CONTINUE HANDLER FOR NOT FOUND SET selesai = 1;
407
408     OPEN d_kursor;
409     model_xmr LOOP
410     fetch d_kursor INTO np,totalp;
411     if selesai = 1 then
412         leave model_xmr;
413     end if;
414     update daftarcalonasdos set ranking=ranking,total_nilai_gtotalp where np=np and kidnasmen=kriteria;
415     END LOOP model_xmr;
416     CLOSE d_kursor;
417 end if;
418 DELIMITER ;

```

Gambar 9. Sintaks lengkap *procedure* hitungranking

dari gambar 9 diatas, proses diawali dengan pengambilan data dair tabel xr dan daftarcalonasdos. Data hasil pemanggilan tersebut disimpan pada variabel yang sudah disiapkan (np dan totalp). Dari baris pertama pemanggilan data tersebut dilakukan proses update isi kolom ranking dan total_nilai dari tabel daftarasdos.

3.3 Pengujian *Store Procedure* Penelitian

Pengujian *Stored Procedure* yang dirancang dalam penelitian ini dilakukan menggunakan data uji simulasi atau data bukan sebenarnya. Pendekatan ini dilakukan untuk memastikan bahwa prosedur dan fungsi yang dibangun dapat berjalan sesuai logika algoritma *Simple Additive Weighting* (SAW). Untuk tujuan yang lain mengapa menggunakan data simulasi pada pengujian *Store Procedure* adalah untuk meminimalkan risiko kesalahan atau inkonsistensi selama tahap

pengembangan dan pengujian. Dengan data yang tidak sensitif dan bebas dari identitas pribadi mahasiswa maupun dosen, proses *debugging* dan evaluasi hasil dapat dilakukan lebih fleksibel. Meskipun menggunakan data simulasi, hasil pengujian memberikan gambaran yang jelas mengenai kinerja dan keakuratan *Stored Procedure* yang dirancang. Akan diambil satu calon asdos dari tabel 3 dan akan dihitung secara manual. Calon asisten dosen yang terpilih untuk pengujian *Store Procedure* adalah calon asdos yang bernama “Nayaka Anung”. Pada pengujian *Store Procedure* penelitian ini akan dilakukan pengukuran akurasi *Store Procedure*, dimana hasil perhitungan *Store Procedure* harus sama dengan hitungan manual. Sebelum dilakukan perhitungan manual maka diperlukan dicari terlebih dahulu nilai minimal dan maksimal dari nilai kriteria asisten dosen. Data uji pada penelitian disajikan pada tabel 3.

Tabel 3. Data uji penelitian

NPM	Nama	CRT 1	CRT 2	CRT 3	CRT 4	CRT 5	CRT 6	CRT 7	CRT 8	CRT 9	CRT 10	CRT 11	CRT 12	CRT 13
		1	2	3	4	5	6	7	8	9	10	11	12	13
021	Muruhita Lathiif	2.89	2.23	4.45	2.1	4.7	3.34	3	2.45	3.1	1.9	4.01	1.89	1.99
344	Nayaka Anung	3.49	1.78	2.54	3.01	4.6	1.76	3.01	2.1	1.01	4.01	2.78	4.9	3.1
354	Yudi Rohman	3.20	4.3	2.1	4.2	1.56	2.21	4.10	2.33	2.11	3.23	3.09	1.09	3.03
Nilai Minimal		2.89	1.78	2.1	2.1	1.56	1.76	3	2.1	1.01	1.9	2.78	1.09	1.99
Nilai Maksimal		3.49	4.3	4.45	4.2	4.7	3.34	4.10	2.45	2.11	4.01	3.09	4.9	3.1

Setelah diketahui nilai minimal dan maksimal dari setiap nilai kriteria asisten dosen, selanjutnya adalah dilakukan proses normalisasi nilai kriteria (nilai R) asisten dosen atas nama Nayaka Anung. Untuk mencari nilai R, harus dilihat terlebih dahulu nilai tersebut berada pada kriteria “Benefit” atau “Cost”. Jika nilai tersebut berada pada jenis kriteria “Benefit”, maka rumus yang digunakan untuk melakukan normalisasi adalah:

$$R_{ij} = \frac{\min(x_{ij})}{x_{ij}} \tag{1}$$

Dan jika nilai tersebut berada pada jenis kriteria “Benefit”, maka rumus yang digunakan untuk melakukan normalisasi adalah:

$$R_{ij} = \frac{x_{ij}}{\max(x_{ij})} \tag{2}$$

Dengan menggunakan rumus (1) dan (2), maka nilai R/nilai yang ternormalisasi calon asisten dosen atas nama “Nayaka Anung” disajikan pada tabel 4.

Tabel 4. Tabel hasil perhitungan nilai R salah satu calon asisten dosen

No	Kode Kriteria	Nilai Minimal kriteria asisten dosen	Nilai maksimum kriteria asisten dosen	Nilai Kriteria Nayaka Anung	Nilai R
1	CRT1	2.89	3.49	3.49	1
2	CRT2	1.78	4.3	1.78	0.413953488
3	CRT3	2.1	4.45	2.54	0.570786517
4	CRT4	2.1	4.2	3.01	0.716666667
5	CRT5	1.56	4.7	4.6	0.978723404
6	CRT6	1.76	3.34	1.76	0.526946108
7	CRT7	3	4.1	3.01	0.734146341
8	CRT8	2.1	2.45	2.1	0.857142857
9	CRT9	1.01	3.1	1.01	0.325806452
10	CRT10	1.9	4.01	4.01	1
11	CRT11	2.78	4.01	2.78	0.693266833
12	CRT12	1.09	4.9	4.9	0.22244898
13	CRT13	1.99	3.1	3.1	0.641935484

Setelah diketahui nilai R calon asisten dosen atas nama Nayaka Anung, selanjutnya dihitung untuk nilai preferensinya dengan menggunakan rumus:

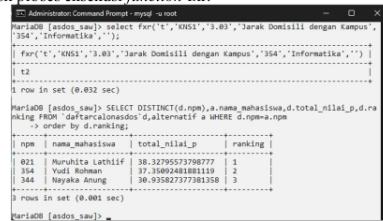
$$P_i = \sum_{j=1}^m w_j r_{ij} \tag{3}$$

Dimana w adalah bobot kriteria (tabel 1). Dengan menggunakan rumus (3), maka perhitungan untuk nilai preferensi calon asisten dosen atas nama Nayaka Anung adalah:

$$P_i = 2.6 * 1 + 4 * 0.413953488 + 4.1 * 0.570786517 + 4.5 * 0.716666667 + 2.9 * 0.978723404 + 3.8 * 0.526946108 + 3.3 * 0.734146341 + 4.7 * 0.857142857 + 4.3 * 0.325806452 + 3.6 * 1 +$$

$$3.4 * 0.693266833 + 3 * 0.22244898 + 2.8 * 0.641935484 = 30.93582738$$

Setelah diketahui hasilnya, selanjutnya hasil tersebut dibandingkan dengan hasil perhitungan menggunakan *Store Procedure*. Untuk mengetahui hasil perhitungan *Store Procedure*, dilakukan eksekusi *function* *fxr* dan Gambar 10 dibawah ini adalah gambar hasil proses eksekusi *function* *fxr*.



```

MariaDB [sdos_saw] select fxr('t', 'KNS1', '3.03', 'Jarak Domisili dengan Kampus',
'354', 'Informatika', '');
+-----+
| fxr('t', 'KNS1', '3.03', 'Jarak Domisili dengan Kampus', '354', 'Informatika', '') |
+-----+
| t2 |
+-----+
1 row in set (0.032 sec)

MariaDB [sdos_saw] SELECT DISTINCT(d.npm), a.nama_mahasiswa, d.total_nilai_p, d.ra
ting FROM 'sdfercalonsdos' d, alternatif a WHERE d.npm=a.npm
-> order by d.ranking;
+-----+-----+-----+-----+
| npm | nama_mahasiswa | total_nilai_p | ranking |
+-----+-----+-----+-----+
| 021 | Marhita Lailif | 38.3279557798777 | 1 |
| 354 | Yuli Rohman | 37.35092481881110 | 2 |
| 344 | Nayaka Anung | 36.935827377381358 | 3 |
+-----+-----+-----+-----+
3 rows in set (0.002 sec)

```

Gambar 10. Hasil proses eksekusi *function* *fxr*

Didalam algoritma *Simple Additive Weighting* (SAW), penentuan pemenang berdasarkan nilai preferensinya. Dan dari gambar 10 diatas, calon asisten dosen atas nama “Nayaka Anung” memiliki peringkat ke 3 dengan nilai preferensi sebesar 30.935827377381358. Jika kita bandingkan dengan perhitungan manual, hasil keduanya sama. Hal ini membuktikan bahwa *Store Procedure* yang dibangun sudah sesuai dengan harapan.

3.4 Perbandingan penelitian

Terdapat beberapa penelitian pendahulu yang menggunakan algoritma *Simple Additive Weighting* (SAW) untuk penentuan dosen dengan menggunakan aplikasi komputer atau kasus multikriteria yang lain. Penelitian tersebut adalah Penelitian Triana Elizabeth dkk (2020)[17], Ramos Somya dkk (2019)[19], Jordy Irawan Antono dkk (2022)[20], Hamid Muhammad Jumasa dkk (2022)[21], Heri Setiawan dkk (2022)[22], dan Nadia Salsabila dkk (2024)[23]. Pada penelitian terdahulu tersebut, algoritma *Simple Additive Weighting* (SAW) diletakkan pada sisi aplikasi, sehingga kelemahannya berupa redundansi kode, keterbatasan efisiensi, risiko inkonsistensi data, serta kerentanan terhadap serangan *SQL Injection*. Penelitian ini algoritma diletakkan pada *Store Procedure* basisdata, sehingga lebih aman dari permasalahan yang dialami oleh penelitian terdahulu.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa perancangan *Store Procedure* pada basis data MySQL untuk mengimplementasikan algoritma *Simple Additive Weighting* (SAW) berhasil diwujudkan secara fungsional dan sesuai dengan tujuan penelitian. Rancangan yang dibuat terbukti mampu menjalankan tiga proses utama dalam SAW, yaitu normalisasi, pembobotan, dan perankingan, secara otomatis melalui *function* dan *procedure*. Pengujian dengan data simulasi menunjukkan bahwa hasil perhitungan preferensi menggunakan *Stored Procedure* identik dengan hasil perhitungan manual. Contohnya pada pengujian dengan calon asisten dosen “Nayaka Anung”, nilai preferensi yang dihitung manual sebesar 30,9358 sama persis dengan hasil eksekusi *Store Procedure*. Kesamaan hasil ini membuktikan keakuratan logika *Simple Additive Weighting* (SAW) yang ditanamkan di dalam basis data. Selain itu, hasil pengujian juga memperlihatkan efisiensi proses, karena seluruh perhitungan dilakukan langsung pada level basis data tanpa redundansi kode di sisi aplikasi.

Dan jika dibandingkan dengan penelitian terdahulu yang menempatkan algoritma SAW di tingkat aplikasi, penelitian ini memberikan keunggulan nyata berupa konsistensi data, serta penguatan aspek keamanan karena meminimalkan risiko *SQL Injection*. Dengan demikian, penelitian ini tidak hanya berhasil mengotomatisasi proses seleksi asisten dosen berbasis multi-kriteria, tetapi juga memberikan kontribusi kebaruan berupa integrasi penuh algoritma SAW di *Stored Procedure*. Hasil ini dapat dijadikan dasar pengembangan lebih lanjut untuk integrasi dengan aplikasi web maupun desktop, sehingga mampu memberikan solusi praktis, objektif, dan dapat diandalkan dalam mendukung pengambilan keputusan akademik.

UCAPAN TERIMAKASIH

Ucapan terima kasih yang tulus penulis sampaikan kepada pimpinan Universitas PGRI Yogyakarta (UPY) yang telah memberikan dukungan, arahan, dan fasilitas selama proses pelaksanaan penelitian ini. Penghargaan yang sama juga penulis tujukan kepada Ketua Program Studi Informatika UPY beserta seluruh jajaran dosen dan staf yang telah

memberikan bimbingan, masukan, serta lingkungan akademik yang kondusif sehingga penelitian ini dapat berjalan dengan lancar.

ORIGINALITY REPORT

8%

SIMILARITY INDEX

7%

INTERNET SOURCES

4%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1	pl.scribd.com Internet Source	1 %
2	Irwin Supriadi. "Implementasi Algoritma SAW Pada Aplikasi Penentuan Pembimbing Tugas Akhir Mahasiswa (Studi Kasus: Prodi Informatika Universitas Langlangbuana)", INTERNAL (Information System Journal), 2021 Publication	1 %
3	jurnal.goretanpena.com Internet Source	1 %
4	docplayer.info Internet Source	<1 %
5	id.scribd.com Internet Source	<1 %
6	repository.its.ac.id Internet Source	<1 %
7	journal.umpalangkaraya.ac.id Internet Source	<1 %
8	jurnal.umnu.ac.id Internet Source	<1 %
9	klik.ulm.ac.id Internet Source	<1 %
10	ripuc.ri.gov Internet Source	<1 %
11	Hidayanti Murtina, Nunung Hidayatun, Susafaati Susafaati. "Pengujian Tingkat	<1 %

Reliabilitas Metode Weight Aggregated Sum Product Assessment Menggunakan Pearson Correlation", Jurnal Teknik Komputer, 2021

Publication

12	proceeding.unisba.ac.id Internet Source	<1 %
13	www16.us.archive.org Internet Source	<1 %
14	text-id.123dok.com Internet Source	<1 %
15	123dok.com Internet Source	<1 %
16	digilib.unila.ac.id Internet Source	<1 %
17	eprints.undip.ac.id Internet Source	<1 %
18	jim.teknokrat.ac.id Internet Source	<1 %
19	ojs.ikipmataram.ac.id Internet Source	<1 %
20	repository.usd.ac.id Internet Source	<1 %
21	Rabiatul Adawiyah, Susilahudin Putrawangsa. "Studi Kompetensi Pedagogik Mahasiswa Calon Guru dalam Merancang Pembelajaran Berparadigma Konstruktivis [Pedagogical Competency Studies of Prospective Teachers in Designing Constructivist Paradigma Learning]", PEDAGOGIA: Jurnal Pendidikan, 2019 Publication	<1 %
22	www.slideshare.net Internet Source	<1 %

23 Nurul Huda, Muhammad Hasbi, Teguh Susyanto. "Seleksi Penerima Bantuan Pangan Non Tunai di Desa Menggunakan Metode Naïve Bayes dan Simple Additive Weighting", Jurnal Ilmiah SINUS, 2021
Publication

24 Yudi Setiawan, Andang Wijanarko, Helmizar. "Analisis Penerapan Protokol Covid-19 Pada Optimasi Distribusi Ruang Pembelajaran Resource Sharing Dengan Implementasi Algoritma Simple Additive Weighting (SAW)", Electrician, 2022
Publication

25 de.scribd.com
Internet Source

26 mafiadoc.com
Internet Source

27 matriks.sipil.ft.uns.ac.id
Internet Source

28 repositori.usu.ac.id
Internet Source

29 www.scribd.com
Internet Source

30 Khoirudin Khoirudin, Muhammad Zakki Abdillah, Saifur Rohman Cholil. "Concentration Determination Disciplines Of Final Projects Using Simple Additive Weighting", Jurnal Transformatika, 2018
Publication

31 Jeffrey Christianto, Rina Candra Noor Santi. "Sistem Pendukung Keputusan Pembelian Laptop Gaming Dengan Metode AHP dan TOPSIS", INTECOMS: Journal of Information Technology and Computer Science, 2024

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off