

Implementasi *Cause-Effect Graphing* pada Pengujian *Black-Box* Aplikasi Kinerja Politeknik Negeri Sriwijaya

Nurul Ilma Hasana Kunio^{1*}, Malahayati², Neneng Miskiyah³, Dzaky Hanifah Ahmad⁴, Siti Kamila⁵

^{1,2,4,5}Jurusan Manajemen Informatika, Politeknik Negeri Sriwijaya, Indonesia

³Jurusan Administrasi Bisnis, Politeknik Negeri Sriwijaya, Indonesia

Email: ¹nurulilmahk@polsri.ac.id, ²malahayati@polsri.ac.id, ³nenengmiskiyah@polsri.ac.id,
⁴dz4kyhanifah@gmail.com, ⁵sitikamila321@gmail.com

INFORMASI ARTIKEL

Histori artikel:

Naskah masuk, 5 Juli 2026

Direvisi, 22 Juli 2026

Diiterima, 31 Juli 2026

ABSTRAK

Abstract- The Sriwijaya State Polytechnic Performance Application was developed to support the integrated management and monitoring of Key Performance Indicator (KPI) data. The complexity of system functions and the numerous combinations of input conditions may lead to errors that affect the accuracy of information. Therefore, software testing is essential to ensure that all system functions operate according to user requirements. This study aims to apply the Cause-Effect Graphing method in black-box testing to validate the relationships between input conditions and system outputs and to generate systematic test scenarios. The research stages consisted of identifying testing requirements, determining causes and effects, constructing a cause-effect graph, developing a decision table, and conducting testing on the application's main features. Testing was performed on two primary features, namely user login and KPI 1 performance calculation, resulting in eight test scenarios derived from the decision table. All test scenarios produced the expected outputs with a 100% success rate, indicating that all business rules of the tested features were successfully validated. The results demonstrate that the Cause-Effect Graphing method can generate systematic test cases, cover various combinations of input conditions, and improve the effectiveness of black-box testing, thereby supporting the software quality of the Sriwijaya State Polytechnic Performance Application.

Kata Kunci:

Black-box testing
Cause-effect graphing
Pengujian perangkat lunak
Aplikasi kinerja

Abstrak- Aplikasi Kinerja Politeknik Negeri Sriwijaya dikembangkan untuk mendukung pengelolaan dan pemantauan data Indikator Kinerja Utama (IKU) secara terintegrasi. Kompleksitas fungsi serta banyaknya kombinasi kondisi masukan berpotensi menimbulkan kesalahan yang dapat memengaruhi keakuratan informasi, sehingga diperlukan pengujian perangkat lunak untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan pengguna. Penelitian ini bertujuan menerapkan metode Cause-Effect Graphing pada pengujian black-box untuk memvalidasi hubungan antara kondisi masukan dan keluaran sistem serta menghasilkan skenario pengujian yang sistematis. Tahapan penelitian meliputi identifikasi kebutuhan pengujian, penentuan cause dan effect, penyusunan cause-effect graph, pembentukan decision table, dan pelaksanaan pengujian pada fitur utama aplikasi. Pengujian dilakukan terhadap dua fitur utama, yaitu login dan perhitungan capaian IKU 1, dengan menghasilkan delapan skenario pengujian berdasarkan decision table. Seluruh skenario menghasilkan keluaran sesuai dengan hasil yang diharapkan dengan tingkat keberhasilan 100%, sehingga seluruh aturan bisnis pada fitur yang diuji berhasil tervalidasi. Hasil penelitian menunjukkan bahwa metode Cause-Effect Graphing mampu menghasilkan kasus uji yang sistematis, mencakup berbagai kombinasi kondisi masukan, serta meningkatkan efektivitas pengujian black-box dalam mendukung kualitas perangkat lunak pada Aplikasi Kinerja Politeknik Negeri Sriwijaya.

Copyright © 2026LPPM - STMIK IKMI Cirebon
This is an open access article under the CC-BY license

Penulis Korespondensi:

Nurul Ilma Hasana Kunio

Program Studi Manajemen Informatika,

Politeknik Negeri Sriwijaya

Jl. Srijaya Negara, Bukit Besar, Palembang, Indonesia

Email: nurulilmahk@polsri.ac.id

1. Pendahuluan

Kinerja perguruan tinggi merupakan salah satu indikator utama dalam menilai keberhasilan penyelenggaraan pendidikan, penelitian, dan pengabdian kepada masyarakat. Dalam era transformasi digital dan tuntutan globalisasi, perguruan tinggi dituntut memiliki sistem tata kelola yang transparan, akuntabel, dan berbasis data guna mendukung pengambilan keputusan strategis. Perkembangan transformasi digital telah mendorong perguruan tinggi untuk mengimplementasikan tata kelola berbasis data (*data-driven governance*) guna meningkatkan transparansi, akuntabilitas, dan efektivitas pengambilan keputusan. Pemanfaatan sistem informasi terintegrasi memungkinkan institusi mengelola data kinerja secara lebih akurat, mempercepat proses monitoring, serta mendukung evaluasi berkelanjutan terhadap capaian organisasi. Digitalisasi tata kelola juga menjadi salah satu faktor penting dalam meningkatkan kualitas manajemen dan daya saing perguruan tinggi di era transformasi digital[1].

Pemerintah Indonesia melalui kebijakan Indikator Kinerja Utama (IKU) mendorong perguruan tinggi untuk melakukan pengukuran kinerja yang tidak hanya berorientasi pada aspek input dan proses, tetapi juga pada outcome dan dampak yang dihasilkan. Politeknik Negeri Sriwijaya (Polsri), sebagai institusi vokasi, dituntut untuk memiliki tata kelola yang transparan dan akuntabel guna memenuhi Indikator Kinerja Utama (IKU) yang ditetapkan oleh Kementerian [2]. Oleh karena itu, pengelolaan data kinerja yang akurat dan terintegrasi menjadi kebutuhan penting bagi setiap perguruan tinggi terutama bagi Polsri.

Pada pelaksanaan perhitungan kinerja perguruan tinggi di Polsri, implementasi pengukuran kinerja masih menghadapi berbagai permasalahan, seperti pengelolaan data yang masih dilakukan menggunakan Microsoft Excel per jurusan, terjadinya duplikasi data, kesalahan input, serta keterlambatan dalam penyajian laporan. Selain itu, keterbatasan integrasi antar jurusan menyebabkan proses pengumpulan dan validasi data menjadi kurang efektif sehingga berdampak pada lambatnya proses evaluasi dan pengambilan keputusan. Untuk mengatasi permasalahan tersebut, Polsri mengembangkan Aplikasi Kinerja berbasis web yang digunakan untuk mendukung pengelolaan dan pemantauan data kinerja secara terintegrasi.

Meskipun aplikasi tersebut telah dikembangkan, kualitas perangkat lunak perlu

dipastikan agar seluruh fungsi yang tersedia dapat berjalan sesuai dengan kebutuhan pengguna. Kompleksitas fitur, banyaknya kombinasi masukan, serta adanya aturan bisnis yang berbeda pada setiap indikator kinerja berpotensi menimbulkan kesalahan fungsi yang dapat memengaruhi keakuratan informasi. Oleh karena itu, pengujian perangkat lunak menjadi salah satu tahapan penting dalam menjamin kualitas aplikasi[3]. Salah satu metode pengujian yang banyak digunakan adalah *black-box testing*, yaitu metode pengujian yang berfokus pada pengujian fungsi sistem tanpa memperhatikan struktur kode program. Salah satu teknik dalam *black-box testing* adalah *Cause-Effect Graphing*, yang digunakan untuk menggambarkan hubungan antara kondisi masukan (*cause*) dan keluaran (*effect*) sehingga menghasilkan kombinasi kasus uji secara sistematis dan efektif [4], [5], [6], [7], [8]. Pendekatan tersebut mampu meningkatkan cakupan pengujian, mengurangi redundansi kasus uji, serta memvalidasi aturan bisnis secara lebih efektif dibandingkan penyusunan kasus uji secara manual[9].

Pada penelitian [10] menunjukkan bahwa metode *Cause-Effect Graphing* dapat membantu dalam menemukan kesalahan fungsi pada sistem agar dapat dilakukan perbaikan sehingga sistem dapat berjalan sesuai dengan fungsinya. Penelitian [10] menerapkan metode ini pada sistem informasi arsip dan menunjukkan bahwa kombinasi *Equivalence Partitioning* dan *Cause-Effect Graphing* mampu meningkatkan kualitas pengujian fungsional. Metode *Cause-Effect Graphing* relatif lebih unggul karena memerhatikan integrasi antar kombinasi input dan output dan dapat mereduksi kasus uji [11].

Hipotesis penelitian ini adalah bahwa penerapan metode *Cause-Effect Graphing* pada pengujian *black-box* mampu menghasilkan kasus uji yang lebih terperinci dalam mengidentifikasi kesalahan fungsi pada Aplikasi Kinerja Polsri [12]. Pendekatan penyelesaian masalah dilakukan melalui tahapan identifikasi kebutuhan pengujian, penentuan *cause* dan *effect*, penyusunan *cause-effect graph*, pembentukan *decision table*, serta pelaksanaan pengujian pada fitur-fitur utama aplikasi.

Berdasarkan kesenjangan tersebut, penelitian ini menerapkan metode *Cause-Effect Graphing* pada pengujian *black-box* Aplikasi Kinerja Politeknik Negeri Sriwijaya. Kontribusi (*novelty*) penelitian ini terletak pada penerapan *Cause-Effect Graphing* untuk memvalidasi aturan bisnis perhitungan IKU melalui kombinasi kondisi masukan yang kompleks sehingga menghasilkan kasus uji yang lebih sistematis, representatif, dan mampu meningkatkan efektivitas pengujian

perangkat lunak. Oleh karena itu, penelitian ini bertujuan mengevaluasi kesesuaian fungsi sistem dengan kebutuhan pengguna, menghasilkan skenario pengujian yang sistematis, serta mendukung peningkatan kualitas perangkat lunak pada Aplikasi Kinerja Politeknik Negeri Sriwijaya.

2. Metode

Penelitian ini menerapkan metode *Research and Development* (R&D) yang dikombinasikan dengan kerangka kerja *Agile Scrum* [13]. Pemilihan metode ini didasarkan pada karakteristik aplikasi kinerja yang dinamis, dimana perubahan indikator atau regulasi sering terjadi sehingga menuntut fleksibilitas yang tinggi dan kecepatan dalam memberikan umpan balik. Pendekatan ini memungkinkan tim beradaptasi dengan perubahan kebutuhan pengguna secara efisien tanpa mengganggu proses pengembangan [14].

Penelitian ini terdiri atas dua aktivitas utama, yaitu pengembangan aplikasi dan evaluasi kualitas perangkat lunak. Metode R&D dan *Agile Scrum* digunakan pada tahap pengembangan aplikasi, sedangkan metode *Cause-Effect Graphing* digunakan sebagai teknik pengujian *black-box* untuk mengevaluasi kualitas fungsi aplikasi yang telah dikembangkan. Dengan demikian, *Agile Scrum* bukan merupakan metode pengujian, melainkan metodologi pengembangan perangkat lunak.



Gambar 1. Alur penelitian

Gambar 1 menunjukkan alur yang digunakan dalam penelitian ini. Berikut penjelasan dari setiap alur yang ada di dalam penelitian ini:

2.1 Identifikasi Masalah

Tahap identifikasi masalah dilakukan untuk mengetahui kondisi serta proses dari pengukuran kinerja di Polsri. Identifikasi dimulai dengan melakukan observasi proses bisnis yang berjalan, wawancara, serta melakukan studi literatur. Hasil identifikasi digunakan sebagai dasar menentukan kebutuhan sistem yang akan dikembangkan [15].

2.2 Analisis Kebutuhan Pengguna

Pada tahap ini dilakukan pengumpulan kebutuhan sistem melalui wawancara, observasi, dan analisis dokumen [16]. Kebutuhan pengguna dibagi dua yaitu kebutuhan fungsional dan nonfungsional.

2.3 Perencanaan

Tahap perencanaan berfungsi untuk menyusun ruang lingkup penelitian, menentukan prioritas fitur yang akan dikembangkan, serta menyusun jadwal pelaksanaan penelitian. Perencanaan dilakukan agar proses pengembangan sistem berjalan secara sistematis sesuai dengan pendekatan Agile.

2.4 Perancangan Fitur

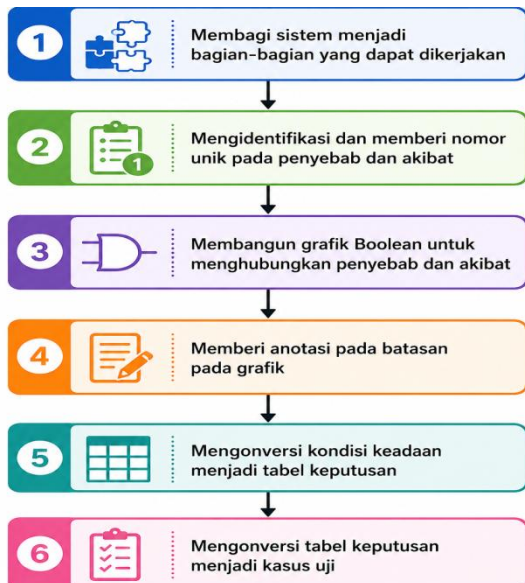
Berdasarkan kebutuhan yang telah diperoleh dari tahapan sebelumnya, dilakukan perancangan sistem yang meliputi perancangan proses bisnis, basis data, antarmuka pengguna, serta rancangan fungsi-fungsi aplikasi yang akan dibangun.

2.5 Pengembangan Sistem

Pada tahapan pengembangan sistem merupakan implementasi rancangan menjadi aplikasi berbasis website. Pengembangan dilakukan secara bertahap sehingga setiap fitur dapat diuji sebelum dilanjutkan ke pengembangan berikutnya.

2.6 Pengujian

Pengujian dilakukan untuk memastikan seluruh fungsi berjalan sesuai kebutuhan pengguna dan spesifikasi sistem. Pada penelitian ini pengujian menggunakan metode *Cause-Effect Graphing*. Metode ini merupakan suatu metode yang memodelkan hubungan logis antara kondisi penyebab (*cause*) berupa masukan atau kondisi tertentu dengan akibat (*effect*) berupa keluaran atau respon yang diharapkan dari sistem [17].



Gambar 2. Tahapan metode cause-effect graphing

Tahapan dari metode *Cause-Effect Graphing* ditunjukkan pada Gambar 2 [12]. Tahap ini dimulai dengan membagi sistem menjadi bagian-bagian yang dapat dikerjakan.

Tahap berikutnya yaitu mengidentifikasi dan memberi nomor unik pada penyebab (*cause*) dan akibat (*effect*). *Cause* merupakan suatu kondisi masukan yang dapat memengaruhi sistem. Sedangkan *Effect* merupakan keluaran yang dihasilkan oleh sistem.

Setelah itu dilakukan pembangunan grafik Boolean untuk menghubungkan penyebab dan akibat. Grafik ini berfungsi untuk menunjukkan keterkaitan antara kondisi masukan dan keluaran yang dihasilkan sistem.

Selanjutnya dilakukan pemberian anotasi pada batasan pada grafik. Kemudian melakukan konversi kondisi keadaan menjadi tabel keputusan. Tahap terakhir yaitu dengan mengonversi tabel keputusan menjadi kasus uji.

Validasi dilakukan dengan membandingkan keluaran aktual sistem terhadap keluaran yang diharapkan pada setiap *test case*. Pengujian dinyatakan berhasil apabila seluruh output sesuai dengan rule pada *decision table*. Persentase keberhasilan dihitung menggunakan persamaan:

$$Success = \frac{Jumlah\ Test\ Case\ Berhasil}{Jumlah\ Test\ Case} \times 100\% \quad (1)$$

2.7 Evaluasi

Tahap evaluasi dilakukan untuk mengetahui kesesuaian aplikasi terhadap kebutuhan pengguna. Masukan yang diperoleh akan menjadi dasar dalam melakukan penyempurnaan aplikasi pada iterasi berikutnya.

3. Hasil dan Pembahasan

Bagian ini menampilkan hasil dan pembahasan Aplikasi Pengukuran Kinerja Polsri dengan menggunakan metode Agile. Hasil penelitian dijelaskan sesuai dengan alur penelitian yang meliputi identifikasi masalah, analisis kebutuhan pengguna, perencanaan, perancangan fitur, pengembangan sistem, pengujian dengan menggunakan metode *Cause-Effect Graphing*, serta evaluasi aplikasi.

3.1 Identifikasi Masalah

Berdasarkan hasil observasi dan wawancara di Polsri, proses pengukuran kinerja masih dilakukan secara terpisah pada masing-masing unit kerja. Pengumpulan data capaian indikator masih banyak menggunakan dokumen spreadsheet sehingga proses rekapitulasi, validasi, dan penyusunan laporan membutuhkan waktu yang relatif lama. Selain itu, pimpinan mengalami kesulitan dalam memantau capaian IKU secara real time.

3.2 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna dilakukan untuk mengidentifikasi kebutuhan fungsional aplikasi berdasarkan tugas dan tanggung jawab setiap pengguna. Hasil analisis menunjukkan bahwa aplikasi memiliki empat jenis pengguna, yaitu Admin, Program Studi (Prodi), Jurusan, dan Pimpinan. Masing-masing pengguna memiliki hak akses dan kebutuhan sistem yang berbeda sehingga fitur yang disediakan disesuaikan dengan peran yang dimiliki. Hasil analisis kebutuhan pengguna dapat dilihat pada Tabel 1.

Tabel 1. Analisis kebutuhan pengguna

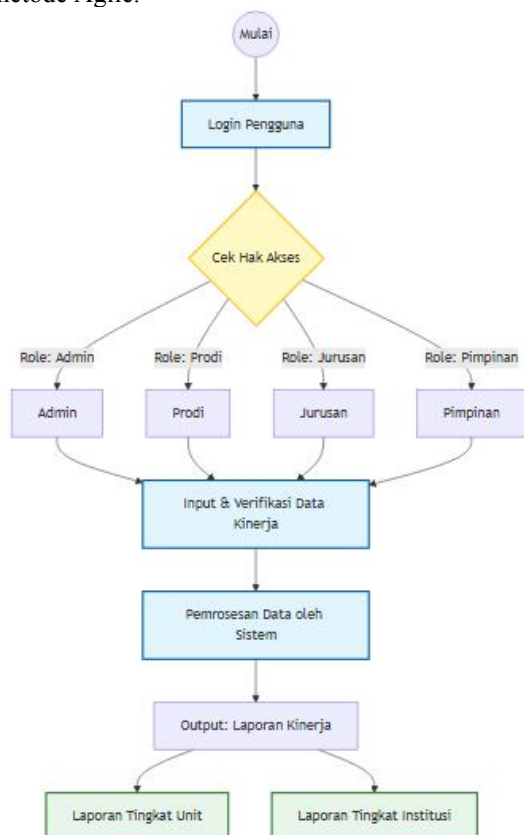
Role	Kebutuhan Pengguna	Fitur Sistem
Admin	a. Mengelola data pengguna, program studi, dan jurusan	a. Tambah, ubah, hapus data pengguna, program studi, jurusan
	b. Mengelola periode triwulan IKU	b. Pengaturan periode triwulan IKU
	c. Melihat rekapitulasi capaian IKU	c. Dashboar rekapitulasi capaian IKU
Prodi	a. Melihat dan menginput data IKU	a. Halaman input capaian IKU
	b. Mengubah data capaian sebelum periode triwulan ditutup	b. Edit dan monitoring data capaian IKU
Jurusan	a. Memantau capaian IKU seluruh prodi di bawah jurusan	a. Dashboard jurusan
		b. Halaman detail analisis capaian IKU

	b. Mengelola detail analisis capaian IKU	
Pimpinan	a. Melihat dashboard capaian IKU keseluruhan b. Melihat data capaian kinerja jurusan	a. Dashboard pimpinan b. Halaman laporan capaian kinerja jurusan

Berdasarkan Tabel 1, setiap role memiliki kebutuhan dan hak akses yang berbeda sesuai dengan tugas dan tanggung jawabnya. Admin berperan mengelola data master, periode triwulan IKU, serta memantau rekapitulasi capaian. Prodi bertugas menginput dan memperbarui data capaian IKU selama periode pengisian berlangsung. Jurusan berfungsi memantau capaian IKU seluruh program studi di bawah koordinasinya serta melakukan analisis terhadap hasil capaian. Sementara itu, Pimpinan menggunakan aplikasi untuk memantau capaian IKU secara menyeluruh dan melihat laporan kinerja sebagai dasar evaluasi dan pengambilan keputusan.

3.3 Perencanaan

Berdasarkan kebutuhan yang telah diperoleh, pengembangan aplikasi dilakukan menggunakan metode Agile.

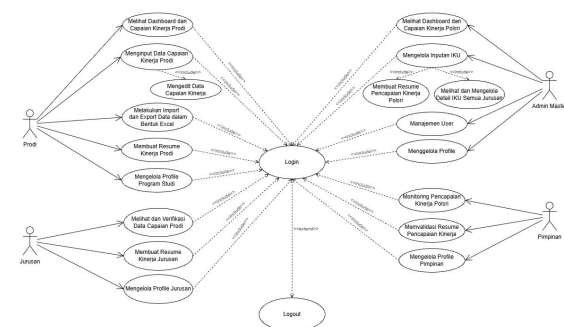


Gambar 3. Alur aplikasi pengukuran kinerja

Gambar 3 menunjukkan alur aplikasi pengukuran kinerja. Pada aplikasi ini fitur-fitur diprioritaskan berdasarkan tingkat urgensi sehingga modul inti, seperti login, pengelolaan IKU, penginputan capaian, dan dashboard monitoring dikembangkan terlebih dahulu sebelum fitur pendukung lainnya.

3.4 Perancangan Fitur

Hasil perancangan menghasilkan model sistem berupa diagram UML, struktur basis data, dan rancangan antarmuka aplikasi. Perancangan ini menjadi acuan dalam proses implementasi sehingga setiap kebutuhan pengguna dapat direalisasikan dalam bentuk fitur aplikasi.



Gambar 4. Use case diagram aplikasi pengukuran kinerja

Gambar 4 menunjukkan use case diagram aplikasi pengukuran kinerja dengan 4 aktor. Aktor Prodi berperan dalam pengelolaan data kinerja pada tingkat program studi. Prodi dapat melihat capaian kinerja prodi, menginput serta mengedit data capaian kinerja, melakukan impor dan ekspor data dalam bentuk file Excel, serta membuat resume kinerja prodi. Aktor Jurusan memiliki fungsi untuk mengawasi dan mengonsolidasikan data kinerja program studi yang berada di bawahnya. Jurusan dapat melihat dan melakukan verifikasi data capaian kinerja prodi, dan membuat resume kinerja jurusan.

Aktor Admin Master bertindak sebagai pengelola utama sistem. Admin Master memiliki kewenangan untuk melihat dashboard dan capaian kinerja institusi, mengelola inputan IKU, membuat resume pencapaian kinerja institusi, melihat dan mengelola detail IKU seluruh jurusan, serta melakukan manajemen pengguna (user management). Sementara itu, aktor Pimpinan berperan dalam pengambilan keputusan berbasis data. Pimpinan dapat memantau (monitoring) pencapaian kinerja institusi, dan memvalidasi resume pencapaian kinerja. Secara keseluruhan, use case diagram ini menggambarkan alur interaksi antara aktor dan sistem yang terintegrasi, dengan fokus pada pengelolaan, pemantauan, serta evaluasi

capaian kinerja secara berjenjang mulai dari tingkat program studi hingga pimpinan institusi.

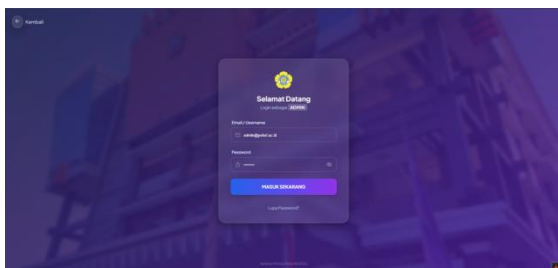
3.5 Pengembangan Sistem

Tahap pengembangan berhasil menghasilkan aplikasi pengukuran kinerja berbasis web yang memiliki fitur autentikasi pengguna, pengelolaan indikator, penginputan target dan realisasi, unggah dokumen pendukung, dashboard capaian, visualisasi grafik, serta laporan otomatis sesuai periode pengukuran..



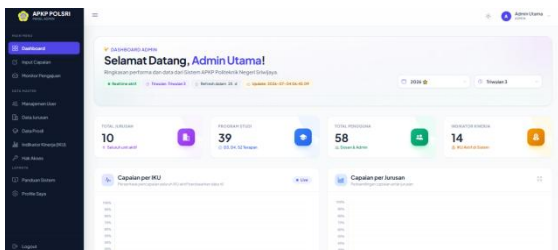
Gambar 5. Tampilan halaman awal

Gambar 5 menunjukkan tampilan halaman awal Aplikasi Pengukuran Kinerja. Halaman tersebut merupakan tampilan halaman awal saat pengguna ingin mengakses sistem. Pada halaman ini, pengguna diminta memilih role untuk login. Role dibedakan menjadi admin, prodi, jurusan, dan pimpinan. Setelah pengguna memilih role akan diarahkan pada halaman login.



Gambar 6. Tampilan halaman login

Gambar 6 menunjukkan tampilan halaman login. Pada halaman ini pengguna diminta untuk memasukkan Email/Username dan Password. Email dan password ini dipergunakan untuk mengakses sistem. Setelah akses berhasil, pengguna dapat mengakses halaman dashboard.



Gambar 7. Tampilan halaman dashboard admin

Gambar 7 menunjukkan tampilan halaman dashboard admin. Pada halaman ini menampilkan informasi seputar pencapaian kinerja dalam bentuk diagram. Pada sistem ini memiliki beberapa modul utama. Setiap modul yang dibuat disesuaikan dengan kebutuhan pengguna, antara lain:

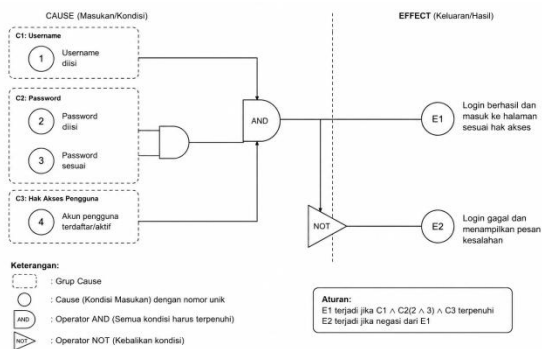
1. Manajemen Pengguna (*User Management*)
Modul ini berfungsi untuk mengatur hak akses berdasarkan peran. Hak akses yang dimiliki antara lain Admin, Program Studi, Jurusan, dan Pimpinan. Hal ini memastikan keamanan dan relevansi data yang diakses oleh setiap unit.
2. Dashboard Monitoring
Modul ini menyajikan visualisasi capaian kinerja secara real-time. Pimpinan dapat melihat persentase ketercapaian target per triwulan melalui grafik yang interaktif, memudahkan pemantauan tanpa harus menunggu laporan akhir tahun.
3. Manajemen IKU
Modul ini digunakan untuk menginput indikator kinerja utama dan tambahan, serta mengunggah bukti dukung capaian. Sistem secara otomatis menghitung nilai kinerja berdasarkan formula yang telah divalidasi.
4. Pelaporan Otomatis
Modul ini digunakan untuk menghasilkan laporan kinerja per unit (Prodi/Jurusan) dalam format PDF siap cetak, yang dapat membantu dalam administrasi pelaporan.

3.6 Pengujian

Pengujian aplikasi dilakukan menggunakan metode *Cause-Effect Graphing* untuk memastikan bahwa setiap fungsi pada aplikasi pengukuran kinerja berjalan sesuai dengan kebutuhan pengguna. Metode ini digunakan untuk mengidentifikasi hubungan antara kondisi masukan (*cause*) dan keluaran (*effect*) sehingga diperoleh skenario pengujian yang mampu memverifikasi logika sistem secara menyeluruh. Pengujian difokuskan pada fitur-fitur utama aplikasi, yaitu proses login serta penginputan data IKU.

a. Pengujian fitur login

Pengujian diawali pada fitur login dengan mengidentifikasi kondisi masukan (*cause*) dan keluaran (*effect*) yang mungkin terjadi. Kondisi masukan meliputi kombinasi username dan password yang valid maupun tidak valid, sedangkan keluaran yang diharapkan berupa keberhasilan login atau pesan kesalahan sesuai kondisi yang diberikan.



Gambar 8. Cause-effect graph fitur login

Berdasarkan grafik Cause-Effect pada Gambar 8, hubungan antara setiap kondisi masukan dan keluaran dapat diidentifikasi sehingga diperoleh kombinasi kondisi yang harus diuji. Grafik tersebut kemudian dikonversi ke dalam decision table untuk menghasilkan skenario pengujian yang lebih terstruktur.

Tabel 2. Decision table fitur login

Kondisi	Rule 1	Rule 2	Rule 3	Rule 4
Cause				
C1 : Username diisi	Y	Y	Y	T
C2 : Password diisi	Y	Y	T	Y
C3 : Password sesuai	Y	T	-	-
C4 : Akun terdaftar	Y	Y	Y	Y
Effect				
E1 : Login berhasil	√	-	-	-
E2 : Login gagal dan menampilkan pesan kesalahan	-	√	√	√

Keterangan :

- Y = ya (kondisi terpenuhi)
- T = tidak (kondisi tidak terpenuhi)
- - = tidak berpengaruh
- √ = output yang dihasilkan

Berdasarkan decision table, diperoleh beberapa test case yang mewakili seluruh kemungkinan kondisi login. Hasil pengujian menunjukkan bahwa sistem berhasil mengarahkan pengguna ke halaman utama apabila username dan password yang dimasukkan benar sesuai hak aksesnya. Sebaliknya, sistem menampilkan pesan kesalahan apabila data autentikasi tidak sesuai.

Tabel 3. Hasil pengujian fitur login

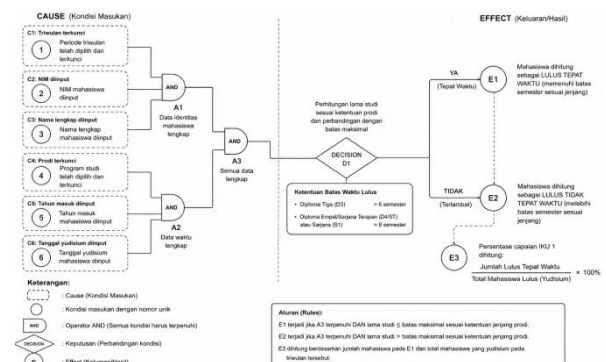
No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Username dan password benar, akun terdaftar	E1 : Login berhasil	E1 : Login berhasil	Berhasil

2	Username benar, password salah, akun terdaftar	E2 : Login gagal dan menampilkan pesan kesalahan	E2 : Login gagal dan menampilkan pesan kesalahan	Berhasil
3	Username benar, password tidak diisi, akun terdaftar	E2 : Login gagal dan menampilkan pesan kesalahan	E2 : Login gagal dan menampilkan pesan kesalahan	Berhasil
4	Username salah, password benar, akun terdaftar	E2 : Login gagal dan menampilkan pesan kesalahan	E2 : Login gagal dan menampilkan pesan kesalahan	Berhasil

Seluruh skenario pengujian pada fitur login menghasilkan keluaran sesuai dengan hasil yang diharapkan sehingga fungsi autentikasi dinyatakan berjalan dengan baik.

Hasil pengujian menunjukkan bahwa *Cause-Effect Graphing* mampu menghasilkan kombinasi kasus uji yang mewakili seluruh kemungkinan kondisi autentikasi. *Cause-Effect Graphing* dapat memodelkan hubungan logika antar kondisi sehingga lebih efektif untuk aplikasi yang memiliki aturan bisnis kompleks.

- Pengujian penginputan data capaian IKU 1
- Pengujian berikutnya dilakukan pada bagian penginputan data capaian IKU 1.



Gambar 9. Cause-effect graph data capaian IKU 1

Gambar 9 menunjukkan Cause-Effect Graph pada proses perhitungan capaian IKU 1, yaitu persentase mahasiswa yang lulus tepat waktu sesuai jenjang studinya. Tahap pengujian diawali dengan mengidentifikasi kondisi masukan (*cause*) yang terdiri atas data triwulan, Nomor Induk Mahasiswa (NIM), nama lengkap, program studi, tahun masuk, dan tanggal yudisium. Seluruh data tersebut menjadi syarat dalam proses perhitungan lama masa studi mahasiswa. Selanjutnya, sistem membandingkan lama masa studi dengan batas maksimal masa studi berdasarkan jenjang

pendidikan, yaitu enam semester untuk Program Diploma Tiga (D3) dan delapan semester untuk Program Diploma Empat/Sarjana Terapan (D4) atau Sarjana (S1). Apabila lama masa studi memenuhi ketentuan tersebut, maka mahasiswa dikategorikan sebagai lulus tepat waktu (E1), sedangkan mahasiswa yang melebihi batas masa studi dikategorikan sebagai lulus tidak tepat waktu (E2). Hasil klasifikasi tersebut kemudian digunakan oleh sistem untuk menghitung persentase capaian IKU 1 (E3), yaitu perbandingan antara jumlah mahasiswa yang lulus tepat waktu dengan total mahasiswa yang lulus pada periode pengukuran, yang selanjutnya ditampilkan pada dashboard sebagai nilai capaian IKU.

Tabel 4. Decision table perhitungan capaian IKU 1

Kondisi	Rule 1	Rule 2	Rule 3	Rule 4
Cause				
C1 : Periode Triwulan	Y	Y	Y	Y
C2 : NIM diisi	Y	Y	T	Y
C3 : Nama mahasiswa diisi	Y	Y	T	Y
C4 : Prodi	Y	Y	Y	Y
C5 : Tahun masuk diisi	Y	Y	Y	T
C6 : Tanggal yudisium diisi	Y	T	Y	T
Decision				
D1 : Lama studi sesuai ketentuan	Y	T	Y	T
Effect				
E1 : Lulus tepat waktu	√	-	-	-
E2 : Lulus tidak tepat waktu	-	√	-	-
E3 : Persentase capaian iku diperbarui	√	-	-	-
E4 : Menampilkan pesan kesalahan	-	-	√	√

Tabel 5. Hasil pengujian perhitungan capaian IKU 1

No	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Seluruh data lengkap dan lama studi sesuai ketentuan	E1 lulus tepat waktu dan E3 diperbarui	E1 lulus tepat waktu dan E3 diperbarui	Berhasil
2	Seluruh data lengkap, namun lama studi melebihi ketentuan	E2 lulus tidak tepat waktu, E3 tidak berubah	E2 lulus tidak tepat waktu, E3 tidak berubah	Berhasil
3	NIM dan nama mahasiswa	E4 sistem menampilkan pesan	E4 sistem menampilkan pesan	Berhasil

4	tidak diisi Tahun masuk dan tanggal yudisium tidak diisi	kesalahan E4 sistem menampilkan pesan kesalahan	kesalahan E4 sistem menampilkan pesan kesalahan	Berhasil
---	--	---	---	----------

Berdasarkan Tabel 4 dan Tabel 5, pengujian perhitungan capaian IKU 1 dilakukan menggunakan empat rule yang merepresentasikan kombinasi kondisi masukan (cause) dan keluaran (effect). Kondisi masukan meliputi periode triwulan (C1), NIM (C2), nama mahasiswa (C3), program studi (C4), tahun masuk (C5), dan tanggal yudisium (C6). Selanjutnya, sistem menghitung lama masa studi (D1) berdasarkan tahun masuk dan tanggal yudisium, kemudian membandingkannya dengan ketentuan masa studi, yaitu maksimal enam semester untuk Diploma Tiga (D3) dan delapan semester untuk Diploma Empat/Sarjana Terapan (D4) atau Sarjana (S1).

Pada Rule 1, seluruh data telah lengkap dan lama masa studi memenuhi ketentuan sehingga mahasiswa dikategorikan lulus tepat waktu (E1) dan persentase capaian IKU 1 diperbarui (E3). Pada Rule 2, seluruh data lengkap, namun lama masa studi melebihi batas ketentuan sehingga mahasiswa dikategorikan lulus tidak tepat waktu (E2) dan tidak menambah persentase capaian IKU 1. Selanjutnya, Rule 3 menunjukkan bahwa data NIM dan nama mahasiswa yang tidak lengkap menyebabkan sistem menampilkan pesan kesalahan (E4). Sementara itu, pada Rule 4, data tahun masuk dan tanggal yudisium tidak diisi sehingga sistem tidak dapat menghitung lama masa studi dan menampilkan pesan kesalahan (E4).

Hasil pengujian menunjukkan bahwa setiap rule menghasilkan keluaran sesuai dengan logika bisnis yang telah dirancang. Sistem mampu memvalidasi kelengkapan data, menentukan status kelulusan tepat waktu, serta menghitung persentase capaian IKU 1 secara otomatis sesuai dengan ketentuan yang berlaku. Hasil penelitian ini sejalan dengan penelitian [10] yang menunjukkan bahwa *Cause-Effect Graphing* dapat meningkatkan cakupan pengujian pada sistem informasi. Namun demikian, penelitian ini memiliki kontribusi tambahan karena menerapkan metode tersebut pada aplikasi pengukuran IKU yang memiliki aturan perhitungan lebih kompleks dibanding

penelitian sebelumnya. Meskipun demikian, metode ini memerlukan identifikasi aturan bisnis yang lengkap. Semakin kompleks hubungan antar kondisi maka proses penyusunan *cause-effect graph* dan *decision table* menjadi lebih sulit.

3.7 Evaluasi

Hasil evaluasi menunjukkan bahwa aplikasi pengukuran kinerja dapat memenuhi kebutuhan pengguna sesuai dengan hasil analisis kebutuhan pada tahap perancangan. Pembagian hak akses berdasarkan peran pengguna, yaitu Admin, Prodi, Jurusan, dan Pimpinan. Pembagian ini dapat membantu pengguna dalam mengakses fitur sesuai dengan tugas dan tanggung jawabnya sehingga proses pengelolaan data menjadi lebih terstruktur.

Aplikasi juga menyediakan fasilitas monitoring melalui dashboard yang menyajikan informasi capaian kinerja secara ringkas dan mudah dipahami. Selain itu, penyajian laporan secara terintegrasi membantu pengguna dalam memperoleh informasi capaian IKU berdasarkan periode pengukuran, sehingga proses monitoring dan evaluasi kinerja dapat dilakukan dengan lebih cepat dibandingkan proses sebelumnya.

4. Kesimpulan

Penelitian ini berhasil menerapkan metode *Cause-Effect Graphing* pada pengujian *black-box* Aplikasi Kinerja Politeknik Negeri Sriwijaya untuk memvalidasi fungsi sistem dan aturan bisnis pada fitur utama aplikasi. Pengujian dilakukan terhadap dua fitur utama, yaitu autentikasi pengguna dan perhitungan capaian IKU 1, dengan menghasilkan delapan skenario pengujian yang disusun berdasarkan *cause-effect graph* dan *decision table*. Seluruh skenario pengujian menghasilkan keluaran yang sesuai dengan hasil yang diharapkan dengan tingkat keberhasilan 100%, sehingga menunjukkan bahwa metode *Cause-Effect Graphing* dapat menghasilkan kasus uji yang sistematis, merepresentasikan berbagai kombinasi kondisi masukan, serta memvalidasi aturan bisnis aplikasi secara efektif.

Hasil penelitian juga menunjukkan bahwa aplikasi yang dikembangkan mampu mendukung proses pengelolaan, pemantauan, dan pelaporan data Indikator Kinerja Utama (IKU) secara lebih terintegrasi, terstruktur, dan efisien. Dengan demikian, penerapan metode *Cause-Effect Graphing* tidak hanya meningkatkan kualitas proses pengujian perangkat lunak, tetapi juga mendukung keandalan aplikasi sebagai sarana pengelolaan kinerja di Politeknik Negeri Sriwijaya.

Penelitian ini masih memiliki keterbatasan karena pengujian hanya dilakukan pada dua fitur

utama aplikasi, sehingga belum mencakup seluruh modul, seperti dashboard, pelaporan, serta proses impor dan ekspor data. Oleh karena itu, penelitian selanjutnya dapat memperluas cakupan pengujian pada seluruh modul aplikasi serta mengombinasikan *Cause-Effect Graphing* dengan teknik *black-box testing* lainnya, seperti *Boundary Value Analysis*, *Equivalence Partitioning*, atau *Decision Table Testing*, untuk memperoleh cakupan pengujian yang lebih komprehensif dan meningkatkan efektivitas validasi perangkat lunak.

Ucapan Terima kasih

Penulis mengucapkan terima kasih kepada Politeknik Negeri Sriwijaya atas dukungan yang diberikan selama penelitian. Penulis juga menyampaikan apresiasi kepada Muhammad Bahrudin dan Helza Framesha selaku tim penyusun dan pengembang aplikasi kinerja Politeknik Negeri Sriwijaya atas bantuan, masukan, dan kerja sama yang telah diberikan selama penelitian ini.

Daftar Pustaka

- [1] J. Komljenovic *et al.*, "Digitalised higher education: key developments, questions, and concerns," *Discourse: Studies in the Cultural Politics of Education*, vol. 46, no. 2, pp. 276–292, Mar. 2025, doi: 10.1080/01596306.2024.2408397.
- [2] Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan, *Buku Panduan Indikator Kinerja Utama Perguruan tinggi Negeri*. Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan, 2024. [Online]. Available: <https://simbelmawa.kemdikbud.go.id/portal/wp-content/uploads/2024/12/Buku-Panduan-Indikator-Kinerja-Utama-PTN-versi2-Kemdikbud-2021.pdf>
- [3] V. Y. P. Ardhana, D. W. Rahma, H. M. Baun, D. Satrinia, and I. S. Widiati, *Rekayasa Perangkat Lunak: Teori dan Konsep*. Sumedang: CV. Mega Press Nusantara, 2025.
- [4] H. Nurfauziah and I. Jamaliyah, "Perbandingan Metode Testing antara Blackbox dengan Whitebox pada Sebuah Sistem Informasi," *Jurnal Visualika*, vol. 8(2), pp. 105–113, 2022.
- [5] A. S. Wati, G. Agiyani, and M. Malahayati, "Cause Effect Graph dan Functional Testing dalam Pengujian Sistem Informasi Perjalanan Dinas," in *Prosiding Seminar Nasional Sains dan Teknologi Terapan*, 2022.
- [6] E. Krupalija, E. Cogo, Š. Bećirović, I. Prazina, and I. Bešić, "Cause-effect Graphing Technique: A Survey of Available Approaches and Algorithms," in *2022 IEEE/ACIS 23rd International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Taichung, Taiwan: IEEE, Dec. 2022, pp. 162–167. doi: 10.1109/SNPD54884.2022.10051799.
- [7] A. Paradkar, K. C. Tai, and M. A. Vouk, "Specification-based testing using cause-effect graphs," *Annals of Software Engineering*, vol. 4, no.

- 1, pp. 133–157, Jan. 1997, doi: 10.1023/A:1018979130614.
- [8] D. Kavzak Ufuktepe, T. Ayav, and F. Belli, “Test input generation from cause–effect graphs,” *Software Qual J*, vol. 29, no. 4, pp. 733–782, Dec. 2021, doi: 10.1007/s11219-021-09560-3.
- [9] J. Weidlich, B. Hicks, and H. Drachsler, “Causal reasoning with causal graphs in educational technology research,” *Education Tech Research Dev*, vol. 72, no. 5, pp. 2499–2517, Oct. 2024, doi: 10.1007/s11423-023-10241-0.
- [10] S. Ismiati, F. P. Aditiawan, and A. L. Nurlaili, “Black Box Testing with the Equivalence Partitioning and Cause Effect Graph Method in Archive Information System,” *J. Tek. Inform. (JUTIF)*, vol. 5, no. 4, pp. 981–990, Jul. 2024, doi: 10.52436/1.jutif.2024.5.4.1944.
- [11] S. Karuniawati, S. Widowati, and I. L. Hakim, “Implementasi Metode Cause Effect Graphing (CEG) dalam Pengujian Requirement Perangkat Lunak (Studi Kasus: Aplikasi G-College),” *e-Proceeding of Engineering*, vol. 2(2), pp. 6475–6480, 2015.
- [12] B. Bekiroglu, “A Cause-Effect Graph Software Testing Tool,” *European Journal of Computer Science and Information Technology*, vol. 5(4), pp. 11–24, 2017.
- [13] L. Judijanto, M. Muhammadiyah, R. N. Utami, L. Suhirman, L. Laka, and Y. Boari, *Metodologi Research and Development (Teori dan Penerapan Metodologi RnD)*. Jambi: Pt. Sonpedia Publishing Indonesia, 2024.
- [14] N. S. Sukmadinata, *Metode Penelitian Pendidikan*. Bandung: Remaja Rosdakarya, 2017.
- [15] N. Y. Arifin, *Dasar Teknologi Informasi*. Jambi: PT Sonpedia Publishing Indonesia, 2025.
- [16] M. A. S. Ekowati, *Analisa dan Perancangan Sistem Informasi*. Bekasi: Penerbit Alungcipta, 2025.
- [17] D. J. Mosley and B. A. Posey, *Just Enough Software Test Automation*. New Jersey: Pearson Education, Inc, 2002.