

Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale

by John Doe

Submission date: 19-Nov-2020 12:52PM (UTC+0700)

Submission ID: 1336497912

File name: ion_Testing_Tool_Dalam_Pengujian_Aplikasi_The_Point_Of_Sale.docx (256.31K)

Word count: 2806

Character count: 17418

Automation Testing Tool Dalam Pengujian Aplikasi

The Point Of Sale

Abstract—Perekaman dan pemutaran kasus uji, normalnya diimplementasikan sebagai test case recording, test case recording tujuannya untuk menampilkan test script yang direkam di jendela kasus uji dan menampilkan perintah dan parameternya dalam format berbentuk tabel. Test case recording telah banyak diterapkan di industry, Salah satu topik utama pengujian otomatis adalah bagaimana menentukan apakah suatu halaman web telah memasuki status siap sebelum perekaman dapat melakukan perintah pengujian selanjutnya? jika elemen web salah dipilih selama pemutaran, maka perintah tes terkait akan maka gagal diproses. penelitian ini mengemukakan mekanisme automatic waiting untuk memainkan perintah tes. Dan hasil percobaan menunjukan bahwa waktu tunggu dapat ditentukan secara otomatis dan dinamis sehingga penguji tidak harus menambahkan perintah tunggu secara manual sehingga mengurangi waktu dan kesalahan buatan. mekanisme yang diusulkan diimplementasikan sebagai bagian dari perangkat lunak pengujian web open source yaitu katalon studio. Katalon studio adalah sebuah software auto test pengujian yang memanfaatkan mesin inti dari selenium yang menawarkan serangkaian fitur-fitur komprehensif untuk mengimplementasikan sebuah solusi pengujian otomatis secara penuh untuk Web,API,dan Mobile.

Keywords—Katalon Studio, Selenium

I. PENDAHULUAN

TPOS merupakan aplikasi website untuk mengelola penjualan produk pada tiap-tiap titik toko di bermacam daerah di Indonesia. Secara teori POS (*Point of sale*) merupakan tempat di mana pelanggan melakukan pembayaran untuk barang atau jasa, dan di mana pajak penjualan dapat dibayarkan. Hal itu bisa dilakukan di tempat toko fisik, di mana terminal POS dan sistem digunakan untuk memproses pembayaran kartu, atau titik penjualan virtual, seperti komputer atau perangkat elektronik seluler. TPOS diharapkan bisa membantu transaksi penjualan dan perhitungan stok barang. Para admin toko bisa melakukan penjualan dan perhitungan stok barang melalui TPOS dari gawai mereka. Manfaat lain yang diharapkan adalah kemudahan admin untuk mengelola toko diberbagai titik daerah di Indonesia melalui satu aplikasi mulai dari informasi penjualan hingga stok produk.

Fokus pembahasan karya ilmiah ini adalah katalon studio pengujian aplikasi web otomatis merupakan hal yang penting topik penelitian dalam rekayasa perangkat lunak. Itu juga dianggap sebagai salah satu jenis pengujian GUI. Masalah utama yang harus diatasi adalah bagaimana caranya mengotomatiskan pembuatan test case.

II. KAJIAN PUSTAKA

A. Katalon Studio

Untuk kebutuhan proses pengujian dilakukan dengan menggunakan tools yaitu Katalon Studio. Tools tersebut merupakan aplikasi open source untuk melakukan pengujian

secara otomatis yang telah dikembangkan oleh Katalon LLC yang dapat dijalankan di semua sistem operasi seperti Windows, MAC os dan linux. Katalon Studio menggunakan antarmuka IDE khusus untuk melakukan pengujian. Katalon Studio memiliki tiga fitur utama untuk menunjang beberapa pengujian dibebberapa platform seperti Web testing, API testing, dan Mobile testing. Selain itu Katalon studio telah terintegrasi dengan beberapa teknologi dari luar seperti github. Katalon Studio sendiri telah menyediakan UI grafis yang beragam tampilan, menu, pohon tabel dan lain-lain untuk mengelola test case objek dan file data. Walaupun masih dalam tahap pengembangan Katalon Studio juga sudah mendukung beberapa environment seperti browser dan sistem operasi komposisinya sangat ideal untuk penguji yang hanya ingin melakukan drag-drop dan memiliki kemampuan pemrograman terbatas.

Meski masih dalam tahap pengembangan katalon juga telah menunjang sebagian *environment* seperti sistem operasi dan browser. Dalam pembuatan test case, katalon mempunyai 3 metode yaitu merekam aktivitas website, membuat manual test case dan membentuk *script*, yaitu *test case* dibuat secara manual oleh penguji

B. Point Of Sale

Sistem POS (*Point of Sale*) ialah suatu sistem untuk menyinkronkan serta mengintegrasikan data pesanan, data reservasi, data kartu hadiah, data e-commerce, ataupun data poin loyalitas yang terletak di fitur POS dengan fitur website orang dagang dan menyinkronkan informasi yang terletak di basis informasi website ke perangkat POS terpaut. Sistem POS digunakan untuk mengintegrasikan sistem *merchant e-commerce* dan sistem pemesanan reservasi, dan lainnya yang disediakan pada situs website. Sitem POS mencakup fitur POS, *server* website POS, susunan basis informasi POS, aplikasi situs website POS.

Adapun publikasi ilmiah yang berjudul *Web Based Point of Sale System* atau disingkat WPOS berbentuk sistem berbasis website yang membolehkan manajemen laporan toko secara jarak jauh, serta membolehkan pelanggan melakukan penjadwalan ataupun penjadwalan ulang waktu pengiriman. Server menyediakan seluruh informasi dan data penting yang dibutuhkan melalui website browser pelanggan. dalam sistem ini, server antar toko bisa berkomunikasi satu sama lain dengan *mainframe* kantor pusat. WPOS bisa diimplementasikan bagaikan rangkaian terintegrasi untuk kerja sama antar posisi toko.

III. METODOLOGI

Pengembangan TPOS menggunakan metode waterfall. Berikut merupakan beberapa penjelasan secara rinci tahap pengembangan aplikasi :

- *Requirements analysis*: Di tahap ini proses menganalisis kebutuhan dan batasan yang dikembangkan

- *System design* : setelah *Requirements analysis* diketahui apa saja kebutuhannya selanjutnya adalah mendesain system. Berfokus pada rancangan antarmuka, fungsi program, algoritma, prosedur, dan arsitektur
- *Implementation*: adalah proses mengubah rancangan yang telah ditetapkan menjadi suatu kode program
- *Integration and Testing*: Tahap ini semua digabungkan dan dilakukan pengujian apakah fungsi program berjalan sesuai apa yang sudah ditentukan
- *Operation and Maintenance*: Tahap ini proses pemeliharaan program seperti perbaikan bug atau meningkatkan sistem TPOS

Pengujian adalah elemen yang paling penting dalam pengembangan perangkat lunak ini juga bisa menjadi aktivitas kompleks untuk menyusun dengan benar, dan dengan cara yang mendukung efisiensi maksimum. Untuk itu tidak ada salahnya tau mengenai 7 principal of software tester, Apa itu 7 principal of software tester atau biasa disebut 7 prinsip pengujian perangkat lunak? Berikut adalah penjelasan mengenai tujuh prinsip pengujian :

- Testing shows presence of defects atau pengujian menunjukkan cacat dalam melakukan pengujian perangkat lunak proses yang dilakukan sebenarnya adalah untuk mencari cacat pada sebuah perangkat lunak yang dikembangkan, yang dimana tester harus membuat sebuah perangkat lunak itu menjadi gagal. Itu diperlukan supaya proses pengujian perangkat lunak bisa menemukan kegagalan program sebanyak banyaknya. Meskipun dalam beberapa proses pengujian sangat mustahil untuk memastikan bahwa sebuah perangkat lunak 100% bebas dari cacat/bug, bahkan jika tidak ada cacat yang ditemukan itu bukan bukti kebenaran
- Early testing atau pengujian dilakukan lebih awal pengujian yang dimulai sedini mungkin dalam pengembangan perangkat lunak, sehingga cacat bisa ditemukan oleh tester diawal testing yang nantinya tester bisa merencanakan testing yang akan dibuat itu seperti apa dan ketika menemukan sebuah cacat/bug nantinya tester bisa memberi saran kepada developer
- **Exhaustive testing is not possible** atau mustahil melakukan pengujian secara menyeluruh melakukan pengujian perangkat lunak secara keseluruhan itu sangat tidak mungkin, tidak perlu melakukan testing secara menyeluruh melainkan bisa juga tester set ekspektasi dan berasumsi membuat skenario batas atas dan batas bawah sebuah perangkat lunak dan tidak perlu mengetes semuanya melainkan menggunakan asumsi bahwa tes sudah dilakukan maka itu akan mencakup keseluruhan
- Testing is context dependent atau pengujian bergantung pada konteks pengujian bergantung pada konteks yang pada dasarnya berarti bahwa cara menguji sebuah situs akan berbeda dari cara menguji aplikasi komersial setiap test case yang dibuat tergantung konteksnya seperti contoh semisal test case A tentang login sedangkan test case B tentang logout itu hal yang berbeda dan ekspektasinya juga berbeda baru bisa melakukan testing. Sehingga pada saat proses pengujian pada perangkat lunak harus benar-benar sesuai dengan konteks dari perangkat lunak itu sendiri karena setiap perangkat lunak itu memiliki cara pengujian yang berbeda masing-masing tidak bisa dipukul sama rata
- **Defect clustering** atau pengelompokan cacat/bug pengelompokan cacat sebuah perangkat lunak pada proses pengujian lebih diutamakan melakukan pengujian pada modul atau fungsional program yang kecil, biasanya semakin kecil modulnya maka cacat/bug yang akan ditemukan akan semakin banyak
- **Pesticide paradox** atau paradox pestisida proses pengujian yang dilakukan secara berulang pada akhirnya tidak akan menemukan cacat pada perangkat lunak. tester tidak bisa begitu saja bergantung pada teknik pengujian yang ada maka harus terus menerus memperbaiki metode yang ada untuk membuat pengujian lebih efektif seperti melakukan kesalahan membiarkan test case tidak di update dan menganggap remeh bahwa test case tersebut selalu hijau dan tidak perlu melakukan update test case itu hal yang salah dan tidak boleh dilakukan
- **Absence of error fallacy** atau tidak ada bug merupakan kesalahan dalam pengujian tidak ditemukan adanya cacat/bug maka hal itu perlu dicurigai, bisa saja perangkat lunak yang 99% bebas dari cacat masih tidak dapat digunakan, hal ini bisa terjadi jika sistem diuji secara menyeluruh untuk persyaratan yang salah maka pengujian perangkat lunak tidak hanya menemukan cacat tetapi juga untuk memeriksa bahwa perangkat lunak memenuhi kebutuhan bisnis, sering menemukan bahwa tesnya berhasil tapi tidak sesuai dengan fungsi jadi ketika error itu tidak ada bukan berarti aplikasi itu dinyatakan bebas lolos dari QA

Kapan menggunakan otomatis testing dan kapan menggunakan manual testing? berikut adalah waktu yang tepat kapan tester harus menggunakan otomatis testing dan kapan tester menggunakan manual testing :

1. Otomatis testing

- Tes kritical dari sebuah produk, tester harus membuat scenario testing untuk core functionality, core functionality biasanya banyak berfungsi pada fungsi fungsi lainnya maka dibutuhkan software testing secara repetitive
- Tes yang harus di eksekusi pada setiap build atau rilis, jika aplikasi yang dimiliki memiliki banyak versi dan harus diuji setiap versinya

- Tes yang harus dijalankan pada banyak kombinasi platform dan os version, seperti 1 skenario di eksekusi untuk platform yang berbeda beda seperti android, ios, web, api dan browser yang berbeda seperti chrome atau Mozilla
- Tes yang dieksekusi pada work flow yang sama tapi menggunakan data yang berbeda beda, seperti contoh semisal ada 1 flow yang selalu di ulang ulang tapi inputan datanya berbeda beda
- Tes yang membutuhkan penginputan data secara masiv dan berulang, contohnya seperti pada saat mengisi form terkadang banyak ada yang berulang juga
- Tes yang digunakan pada performance testing, biasanya akan dilakukan pada performance tes pada sebuah aplikasi kalau itu pakainya manual bagaimana caranya kita mengeksekusinya
- Tes yang dalam eksekusinya membutuhkan waktu yang lama, biasanya dilakukan smoke test atau tes yang berjalan terus terusan itu harus menggunakan automted tes

2. Manual manual testing

- Tes yang hanya dieksekusi 1 kali
- User experience, tes yang membutuhkan user untuk merasakan secara langsung atau mendapat eksperience secara langsung terhadap sebuah alikasi atau real user
- Tes yang harus dilakukan secara emergency, kalau ada bug fix yang harus segera di tes dan kalau menggunakan automation script biasanya akan membutuhkan waktu sedikit lebih lama dibanding manual tes
- Tes yang dilakukan secara random / exploratory testing, ada kalanya sebagai software tester harus bermindset testing shows presence of defects jadi harus explore terus mencari corner case yang belum kita tes
- Tes yang membutuhkan konfirmasi visual, contoh semisal yang ingin pixel perfect buttonya ditengah tengah atau kurang marginya ada ditengah kanan sedikit kiri sedikit terlalu kepinggir 1pixel 2pixel atau gradien warnanya

Kesimpulan tergantung dari kebutuhan testing yang dilakukan jika membutuhkan pengujian yang repetitif atau harus menguji banyak platform data yang besar sebaiknya itu di buat automation script dan jika sifatnya cuma 1 kali testing atau dibutuhkan perasaan atau eksperience langsung biasanya sifatnya emergency sebaiknya manual juga bisa dipertimbangkan

IV. HASIL DAN PEMBAHASAN

A. Kebutuhan sistem

Kebutuhan minimum sistem untuk menjalankan katalon studio diperlihatkan pada table 1

Tabel 1. Kebutuhan sistem

	Kebutuhan
OS	Windows 7,8, 10, macOS 10.11+, Linux (Ubuntu)
CPU	Minimum 1 GHz 32-bit (x86) atau 64-bit (x64) processor
Memori	Minimum: 1 GB RAM (32-bit) atau 4 GB RAM (64-bit) Recommend: 4 GB RAM (32-bit) or 8 GB RAM (64-bit)
Hard Drive	minimum 1 GB ruang tersedia. Dibutuhkan ruang tambahan bergantung pada proyek dan laporan eksekusi.

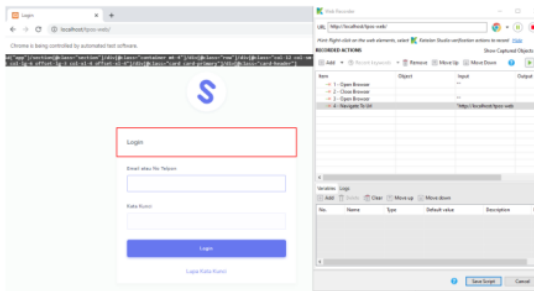
Spesifikasi browser yang mendukung katalon studio diperlihatkan pada table 2. device yang digunakan pada penelitian kali ini, hanya berfokus pada desktop

Table 2. spesifikasi browser

Desktop Browsers	Version on Windows	Version on macOS
Internet Explorer	9, 10, 11	
Microsoft Edge	Current	
Firefox	56+	Untuk Firefox 57, gunakan Katalon Studio v5.1+
Google Chrome	58+	
Opera	Not supported	
Safari	5.1+	9, 10, 11
Opera	Not supported	

B. Proses pengujian dengan katalon

Pada penelitian kali ini yang dilakukan, test case pengujian dibuat dengan metode record & playback yang disediakan oleh katalon. Record/rekam merupakan suatu proses perekaman segala event yang diterima dari seluruh aktivitas pengguna pada halaman web. Seluruh hasil perekaman nanti akan menjadi test yang akan disimpan oleh katalon, dan dijadikan test case. Test case bisa diganti/modifikasi oleh penguji, untuk penyempurnaan proses pengujian, jika dibutuhkan test case tersebut berikutnya dijalankan secara otomatis menggunakan fungsi *playback*. Gambar 1 merupakan proses perekaman pada sebuah halaman web



Gambar 1. proses perekaman event pada katalon

Ketika test case dijalankan, log dari tiap langkah secara otomatis akan terbuat. Log tersebut berisi status status berhasil atau gagalnya suatu langkah dan waktu yang dibutuhkan untuk menyelesaikan langkah tersebut. Ketika terdapat langkah yang gagal maka proses saat menjalankan test case akan berhenti dan rincian test case tersebut bisa dilihat pada log view.

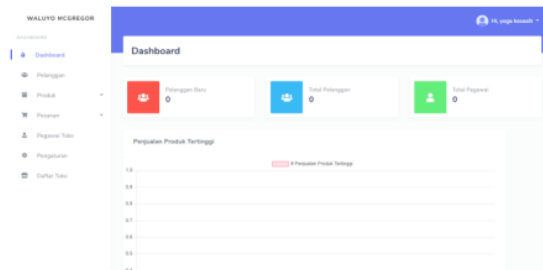
C. Skenario eksperimen

Halaman yang diteliti dari aplikasi web TPOS adalah halaman adalah halaman untuk melihat produk dan pesanan produk karena pemesanan produk merupakan fungsi utama dari aplikasi *e-commerce*. Ada beberapa halaman yang akan diteliti diantaranya :

- Halaman utama
Dihalaman utama hal yang akan diamati adalah waktu membuka web dari TPOS
- Halaman login
Proses akan diamati dari mulai pada saat user memasukan username dan password yang cocok dengan apa yang sudah terdaftar di aplikasi
- Halaman pencarian produk dan pesanan
Pada halaman ini, pengguna akan memakai daftar produk dan pesanan semua proses tersebut diamati
- Halaman rincian produk dan pesanan
Penguji akan memilih produk dan pesanan dan mengamati. Pengamatan rincian produk dan pesanan akan berakhir pada saat pengguna telah masuk halaman rincian produk dan pesanan yang sudah di pilih

Kompleksitas halaman website bisa diukur dari bermacam berbagai aspek seperti browser, banyaknya class, kecepatan data dan lain-lain. TPOS sendiri pasti memiliki kompleksitas yang sangat tinggi. Dalam penelitian kali ini

kompleksitas dapat diukur dengan cara mengamati secara langsung objek yang dimuat dalam suatu halaman



Gambar 2. Dashboard TPOS

Halaman utama aplikasi TPOS memiliki berbagai macam komponen seperti bagian bar samping kiri terdapat dashboard, pelanggan, produk, pesanan, pegawai toko, pengaturan, daftar toko, Dan disamping kanan atas terdapat profil pengguna

D. Hasil

Pengujian dilakukan untuk menguji *respon time* pada aplikasi TPOS dengan menggunakan desktop yang memiliki spesifikasi seperti pada table 3

Table 3. Spesifikasi dekstop

Komponen	Spesifikasi
Sistem Operasi	Windows 10
CPU	2.50GHz 2.71GHz 64-bit (x64) processor
Processor	Intel® Core™ 15-7200 processor
Memori	4 GB RAM
Hard Drive	1 TB
Koneksi Internet	10 Mbps

Pengujian dilakukan dalam beberapa tahapan yang pertama yaitu membuat test case. Test case yang dibuat digunakan untuk menguji respon time dari halaman utama TPOS, halaman login, dashboard, pelanggan, produk, pesanan, pegawai toko, pengaturan, dan daftar toko. Test case untuk pengujian memanfaatkan test case yang dihasilkan dari perekaman tiap event yang dicoba penguji pada halaman tersebut. Test case yang telah dihasilkan lalu dimodifikasi kembali karena terdapat event yang tidak bisa terekam oleh katalon seperti *scroll mouse*.

Tahapan berikutnya adalah test case tadi yang telah dimodifikasi selanjutnya dieksekusi untuk mendapatkan respon time dari tiap-tiap halaman yang diuji. Eksekusi test case di jalankan secara otomatis oleh katalon, test case yang digunakan untuk menguji mulai dari halaman utama sampai dengan pesan produk seperti gambar 3.

Item	Object	Input	Output
-> 1 - Open Browser		***	
-> 2 - Navigate To Url		"http://localhost/tpos-web	
-> 3 - Double Click	input_Email atau No Telpor		
-> 4 - Set Text	input_Email atau No Telpor	"yoga.kosasih172@gmail.c	
-> 5 - Set Encrypted Text	input_Kata Kunci_password	"60M04RNVXgg8T2coUCJ4	
-> 6 - Send Keys	input_Kata Kunci_password	Keys.chord(Keys.ENTER)	
-> 7 - Click	input_Email atau No Telpor		
-> 8 - Double Click	input_Email atau No Telpor		
-> 9 - Set Text	input_Email atau No Telpor	"yoga.kosasih172@gmail.c	
-> 10 - Set Encrypted Text	input_Kata Kunci_password	"o4eAFY7X7Y="	
-> 11 - Send Keys	input_Kata Kunci_password	Keys.chord(Keys.ENTER)	
-> 12 - Click	a_Login		
-> 13 - Click	span_Produk		

Gambar 3. test case untuk menguji TPOS

Pada test case tersebut item 1 sampai 27 berfungsi untuk menguji respon time mulai dari login sampai pemesanan produk.

Gambar 4. Menunjukkan response time yang diperoleh dari pengujian menggunakan katalon



Gambar 4. Hasil pengujian

E. Pembahasan

Dari segi kompleksitas hampir semua memiliki kompleksitas yang seimbang semua memproses suatu proses rata-rata 10 detik. Dalam pengujian dengan menjalankan test case respon tiap test case berbeda-beda apabila dijalankan beberapa kali.

Analisis juga dilakukan terhadap program yang dipakai untuk pengujian ini adalah katalon studio. Katalon memiliki fitur record/merekam agar membantu penggunaanya dalam membuat test case agar lebih mudah dan katalon sangat sensitif terhadap error pada test case terkadang dalam merekam aktivitas untuk test case banyak event yang coba dilakukan oleh user yang tidak berjalan dalam test case atau banyak bagian-bagian yang terekam oleh sendirinya. Maka

test case tersebut perlu diperbaiki secara manual agar test case berjalan dengan semestinya.

V. KESIMPULAN

Di pasar automation tool, selenium kemungkinan besar akan menjadi tools pilihan pertama bagi tester. Selenium berguna untuk menulis test scripting untuk aplikasi web. Selenium mampu melakukan tugas-tugas sederhana memeriksa bagaimana halaman dimuat, ke proses yang lebih kompleks seperti meniru seluruh pengalaman pengguna di situs web. Selama bertahun-tahun selenium telah berkembang menjadi lebih dari sekedar alat otomatisasi untuk keterampilan pengujian yang sangat dibutuhkan bagi para professional quality assurance tetapi di bawah gelombang tren agile dan scrum yang meningkat tim diharapkan mampu melakukan rilis cepat dengan output berkualitas tinggi dengan bantuan alat otomatisasi. Karena pasar lebih membutuhkan tools yang lebih gesit dan komprehensif katalon telah mengungkapkan kelemahan selenium dan keterbatasannya dalam meningkatkan pertumbuhan untuk tim.

Katalon studio dibangun sebagai bundle terpadu yang mencakup hampir semua hal yang diperlukan seperti Java, Android SDK, Web driver. Hasil percobaan menunjukkan bahwa waktu dapat ditentukan secara dinamis dan otomatis, sehingga tester tidak perlu menambahkan secara manual. Mekanisme yang diusulkan adalah diimplementasikan sebagai bagian dari pengujian web *open source* dan telah diadopsi lebih lanjut oleh Katalon Studio dan pada saat ini lebih dari 10.000 tester di seluruh dunia menggunakan mekanisme ini untuk pengujian aplikasi web.

VI. REFERENCES

- [1] C. A. Sukandar, "Warta Ekonomi," 23 April 2019. [Online]. Available: <https://www.wartaekonomi.co.id/read224883/apa-itupoint-of-sale.html>. [Accessed 18 Desember 2019].
- [2] Katalon LLC, "Katalon Documentation", 2018, [Online]. Tersedia : <https://docs.katalon.com/katalonstudio/docs/index.html> [20 juni 2020].
- [3] S. I. O. and V. (. , "Web integrated point-of-sale system". United States Patent US 2012/0296679 A1, 22 November 2012.
- [4] M. Manno, "Web Based Point of Sale System". United States Patent US 2004/0181454 A1, 16 September 2004.

Automation Testing Tool Dalam Pengujian Aplikasi The Point Of Sale

ORIGINALITY REPORT

6%

SIMILARITY INDEX

6%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

PRIMARY SOURCES

1

www.wartaekonomi.co.id

Internet Source

2%

2

docs.katalon.com

Internet Source

2%

3

vizi.blog.binusian.org

Internet Source

1%

4

automated-360.com

Internet Source

1%

5

jurnal.polgan.ac.id

Internet Source

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography On