Penerapan *Metamorphic Testing* pada Aplikasi Tebak Kendaraan

Arlinta Christy Barus[1], Rich Simamora[2], Hanna Tobing[3], Sumiati Hutagalung[4]

[1,2,3,4]Politeknik Informatika Del

[1]arlinta@.del.ac.id, [2]richsimamora@gmail.com, [3]hanna@.del.ac.id, [4]sumiati@.del.ac.id

k

iorphic Testing (MT) adalah sebuah metode pengujian yang melibatkan input, output, dan relasi input dan output suatu program yang dibangkitkan melalui properti yang dimiliki program, yang t dengan *Metamorphic relation* (MR). Relasi pada MR digunakan untuk menemukan *failure* yang da program yang diuji. Dalam pengerjaan tulisan ini, aplikasi yang dijadikan sebagai studi kasus menerapkan MT adalah Aplikasi Tebak Kendaraan. Setelah melakukan pengujian terhadap i tersebut, penulis membuktikan bahwa MT mampu menyingkap *failure* yang ada pada aplikasi tendaraan

unci: *software testing, metamorphic testing (MT), Oracle problem, metamorphic relation* (MR)

ct

orphic Testing (MT) is a software testing method involving inputs, outputs, and the relationships ogram under testing, which are known as *Metamorphic relations* (MRs). The relationships are used ul failures of the program under testing. In this paper, we use an application to determine the type les as a program under testing. From the experiment conducted, we found that the set of MRs de- the study were able to detect some failures of the program under testing.

ds: software testing, metamorphic testing (MT), *Oracle problem, metamorphic relation* (MR)

dahuluan

isan ini berfokus pada pembahasan menge- le problem. Sebuah Oracle pada pengujian at lunak adalah sebuah cara yang digunakan a tester untuk menentukan apakah output gram yang sedang diuji adalah benar [2]. berapa situasi, Oracle tidak tersedia atau ulit untuk menggunakannya. Hal ini dikenal *Oracle problem*"[2]. Situasi yang dimaksud jadi karena kompleksnya sebuah perangkat ng harus diuji atau banyaknya input domain us diuji sehingga tidak mungkin untuk me- pengujian semua input domain terhadap t lunak. Input domain merupakan rentang au bidang masukan untuk sebuah perangkat

toh dari Oracle yang tidak tersedia a ada pada program yang lementasikan fungsi matematik. Fungsi k yang dibuat dalam program terlalu rumit sulit untuk melakukan verifikasi output anual dan sulit untuk menemukan *tools* at mengecek kebenaran output program Kesulitan inilah yang dikenal dengan *oblem*.

morphic Testing (MT) adalah sebuah engujian yang dikembangkan oleh Chen mengurangi *Oracle problem*[3]. Pada MT, berbagai input dan output yang dapat sebagai acuan. Relasi pada MT disebut *hic relation* (MR)[4]. Dengan menggunakan informasi-informasi dari kasus uji yang sudah dieksekusi atau disebut dengan *source test case*, MR digunakan untuk membangkitkan *follow up test case* dan memvalidasi output yang terlibat dari sebuah perangkat lunak[4]. Follow up test case adalah kasus uji selanjutnya yang dibangkitkan dari kasus uji yang telah ada sebelumnya [4].

Pada tulisan ini, aplikasi yang dijadikan sebagai studi kasus untuk diuji dengan MT adalah Aplikasi Tebak Kendaraan yang merupakan aplikasi buatan seorang mahasiswa Politeknik Informatika Del.

Fungsi – fungsi yang terdapat pada studi kasus terdiri dari proses komputasi sederhana namun perlu dibuktikan benar atau tidaknya hasil dari proses komputasi tersebut. Jika pengecekan hasil komputasi dilakukan secara manual maka hal tersebut akan menjadi sulit dan tidak efisien. Hal inilah yang disebut dengan Oracle *problem* yang terdapat pada studi kasus.

Tulisan ini disajikan dalam beberapa bagian sebagai berikut: tinjauan pustaka, deskripsi studi kasus yang akan dikaji, analisis, desain eksperimen, hasil dan pembahasan serta kesimpulan.

2. Tinjauan Pustaka

Ada dua masalah dalam pengujian perangkat lunak, yaitu: Reliable Test Set Problem dan Oracle Problem [1].

yang digunakan untuk menguji fungsi ini relasi antara jari-jari roda dan keliling akan mempengaruhi keliling roda MR yang dihasilkan untuk fungsi ini

Teb1 = jika jari-jariRoda DepanSource jari-jariRodaDepanFollow maka RodaDepanSource > kelilingRoda

jariRodaDepanSource adalah hasil random RodaDepan (20-150) dan jari–jari Roda adalah hasil random jari-dengan kondisi:

Follow-up test case < jari-jariRodaDepan test case

keliling Roda Depan Source > keliling Follow maka tidak ada failure yang terungkap. Namun, jika keliling Roda < keliling Roda Depan Follow maka terungkap.

- fungsi hitungWaktuTempuh(jari-jari)

yang digunakan untuk menguji fungsi ini relasi antara jarak kota tujuan dan waktu yang dibutuhkan kendaraan untuk mencapai tersebut. MR yang dihasilkan untuk adalah:

Teb2 = jika jarakKotaTujuanSource > TujuanFollow maka TempuhSource > WaktuTempuhFollow.

KotaTujuanSource adalah hasil random Tujuan / jarak dari kota asal sampai ke (Laguboti – Medan, Laguboti - Balige, – Porsea, Laguboti – Lumban Lobu, – Parapat, Laguboti – Siantar, Laguboti – Tinggi, Laguboti – Sei Rampah, Laguboti – Morawa) dan JarakKotaTujuanFollow hasil random JarakKotaTujuan dengan

Follow–up test case < JarakKotaTujuan pada

WaktuTempuhSource > WaktuTempuh maka tidak ada failure yang berhasil Namun, jika WaktuTempuhSource < TempuhFollow maka failure berhasil

Teb3 = jika KerusakanJalanSource > KerusakanJalanFollow maka TempuhSource > waktuTempuhFollow.

KotaTujuanSource adalah hasil random Tujuan / jarak dari kota asal sampai ke (Laguboti – Medan, Laguboti - Balige, – Porsea, Laguboti – Lumban Lobu, – Parapat, Laguboti – Siantar, Laguboti – Tinggi, Laguboti – Sei Rampah, Laguboti –

Jika WaktuTempuhSource > WaktuTempuhFollow maka tidak ada failure yang berhasil terungkap. Namun, jika WaktuTempuhSource < WaktuTempuhFollow maka failure berhasil terungkap.

3. Fungsi 3 – fungsi HitungKecepatanTempuh(jari-jari)

MR yang digunakan untuk menguji fungsi ini merupakan relasi antara kecepatan tempuh dan jarak kerusakan jalan. Jarak kerusakan jalan mempengaruhi kecepatan tempuh dari kendaraan tersebut. MR yang dihasilkan untuk fungsi ini adaiah:

MRTeb4 = jika jarak KerusakanJalanSource > jarak KerusakanJalanFollow maka kecepatan TempuhSource < kecepatanTempuhFollow.

JarakKotaTujuanSource adalah hasil random JarakKotaTujuan / jarak dari kota asal sampai ke kota tujuan (Laguboti – Medan, Laguboti - Balige, Laguboti – Porsea, Laguboti – Lumban Lobu, Laguboti – Parapat, Laguboti – Siantar, Laguboti – Tebing Tinggi, Laguboti – Sei Rampah, Laguboti – Tanjung Morawa) dan JarakKotaTujuanFollow adalah hasil random JarakKotaTujuan dengan kondisi:

0 < Follow–up test case < Jarak Kota Tujuan pada source test case

Jika kecepatan Tempuh Source < kecepatan Tempuh Fo!low maka tidak ada failure yang berhasil terungkap. Namun, jika kecepatan Tempuh Source > kecepatan Tempuh Follow maka failure berhasil terungkap.

MRTeb5 = jika jari-jariRodaDepanSource > jari-jariRodaDepanFollow maka kecepatanTempuhSource > kecepatanTempuhFollow.

Jari-jariRodaDepanSource adalah hasil random jari-jariRodaDepan (20 - 150) dan jari-jariRodaDepanFollow adalah hasil random jari-jariRodaDepan dengan kondisi

15 < Follow –up test case < jari-jariRodaDepan pada source test case

Jika kecepatanTempuhSource > kecepatanTempuhFollow maka tidak ada failure yang berhasil terungkap. Namun, jika kecepatanTempuhSource < kecepatanTempuhFollow maka failure berhasil terungkap.

MRTeb6 = jika jarakKerusakanJalanSource > jarakKerusakanJalanFollow maka perlambatanKendaraanSource > perlambatanKendaraanFollow.

Jari-jariRodaDepanSource adalah hasil random JarakKotaTujuan / jarak dari kota asal sampai ke kota tujuan (Laguboti – Medan, Laguboti - Balige,

tersebut [6].

Oracle problem adalah masalah ketika tester sulit untuk menentukan apakah output dari suatu program benar atau tidak [2]. Masalah tersebut diakibatkan kompleksnya input domain yang harus diuji. Input domain adalah semua input yang mungkin digunakan untuk sebuah program.

Test Oracle adalah mekanisme yang dilakukan oleh tester untuk menentukan apakah pada sebuah program terdapat failure atau tidak dengan cara membandingkan output yang diharapkan dengan output yang sebenarnya [9]. Salah satu teknik pengujian perangkat lunak yang mencoba untuk menyelesaikan Oracle problem dengan menerapkan test Oracle adalah *Metamorphic Testing* (MT).

Salah satu contoh dari Oracle problem terjadi pada mesin pencari yaitu Google dengan alamat www.google.com. Pada halaman Google, masukkan input 'ACM *Transactions*'. Sesuai dengan spesifikasi yang ada pada Google, spasi yang diketikkan dikenali sebagai relasi 'AND'. Sistem akan mengembalikan sekitar 8,660,000 hasil untuk pencarian tersebut. Bagaimana tester dapat menentukan output yang benar dan sesuai dengan kata yang dimasukkan oleh user dari hasil yang sangat banyak tersebut? Misalkan lagi, user yang lain memasukkan input 'ACM Transactions "World Wide Web"'. Sesuai dengan spesifikasi pada Google, tanda kutip (" ") berarti pencarian akan mengembalikan hasil yang sesuai dengan kata yang ada di dalam kutip tersebut tanpa dipisah. Sistem akan mengembalikan sekitar 834,000 hasil. Bagaimana tester dapat menentukan bahwa benar – benar hanya 834,000 halaman saja yang relevan dengan input tersebut? Hal inilah yang dikenal sebagai Oracle problem [11].

Metamorphic Testing (MT) adalah sebuah metode pengujian yang dikembangkan oleh Chen untuk mengurangi Oracle problem[3]. Metamorphic Testing merupakan sebuah teknik untuk membangkitkan *follow-up test case* berdasarkan test case yang sudah ada (*source test case*) dan belum menemukan failure. Follow-up test case dibangkitkan dengan merujuk ke "*metamorphic relation*" (MR) [2]. *Metamorphic Testing* melakukan testing dengan cara memverifikasi *metamorphic relation* dari sebuah fungsi yang terdapat pada program yang sedang di uji [10].

Konsep *Metamorphic relation* memiliki beberapa karakteristik berikut ini. Pertama, MR merupakan sebuah properti penting dari fungsi yang akan diuji. Jika salah satu dari properti yang dimiliki oleh fungsi tersebut tidak dipenuhi selama pengujian maka program yang disebut dengan p dapat dinyatakan sebagai program yang salah. Kedua, sebuah *metamorphic relation* merupakan sebuah hubungan/relasi antara input *source* dengan input

maka dapat dikatakan bahwa fungsi
mengandung error.

3. Analisis Studi Kasus

Dalam pengerjaan tulisan ini, apli
akan dijadikan sebagai studi kas
menerapkan MT adalah Aplikasi Tebak K

Aplikasi Tebak Kendaraan adala
sederhana yang berfungsi untuk menghit
perjalanan yang ditempuh dan menge
kendaraan berdasarkan input jari-jari
daerah tujuan. Inputan jari-jari roda ya
ada dua, yaitu jari-jari roda depan dan ja
belakang. Sebenarnya proses komput
terdapat pada aplikasi ini masih sederhan
saat jumlah dan variasi data yang diguna
banyak maka akan tidak efisien apabila t
mengecek output satu per satu secara m
inilah yang menjadi Oracle problem pada
ini.

4. Fungsi Yang Diuji dan MR yang Di

Pada bagian ini dijelaskan mengen
terhadap fungsi – fungsi yang ada pada stu

a. Fungsi HitungKelilingRoda(jari-jari)
   Fungsi ini digunakan untuk menghitu
   roda kendaraan. Keliling roda kendar
   dari dua jenis yaitu keliling roda depa
   belakang. Keliling roda kendaraan
   untuk membuktikan apakah besarnya
   roda mempengaruhi keliling roda terse
b. Fungsi HitungWaktuTempuh(jari-jari)
   Fungsi ini digunakan untuk menghit
   tempuh kendaraan. Waktu tempuh
   dihitung berdasarkan kecepatan dan j
   ditempuh oleh kendaraan.
c. Fungsi HitungKecepatanTempuh(jari-
   Fungsi ini bertujuan untuk m
   kecepatan tempuh kendaraan.
d. Fungsi TentukanJenisKendaraan(jari-j
   Fungsi ini digunakan untuk menent
   kendaraan. Jenis kendaraan terse
   diketahui berdasarkan besarnya jari
   kendaraan.

Dari semua fungsi yang terdapat pa
Tebak Kendaraan ini, fungsi yang akan di
Fungsi HitungKelilingRoda(jari-jari)
Fungsi1, Fungsi HitungWaktuTempu
sebagai Fungsi2,
HitungKecepatanTempuh(jari-jari) sebagai

Pada bagian ini dijelaskan
Metamorphic relation yang dibangkitkan u
kasus Aplikasi Tebak Kendaraan. Berikut
penjelasan dari setiap MR yang terbe
hubungan antara komponen-komponen
Aplikasi Tebak Kendaraan.

1. Fungsi 1 – fungsi hitungKelilingRoda(ja

ISSN:2302-1896

MR yang paling efektif menyingkap failure adalah MRTeb5 karena MR tersebut berhasil menyingkap failure yang lebih banyak dibanding MR yang lain.

Saran

pengujian pada studi kasus eksperimen yang dilakukan, mampu membangkitkan MR ap *failure* yang diterapkan ke

g digunakan untuk membantu mampu menerapkan MR yang lure

lak berhasil disingkap dalam penyebab yang terjadi yaitu tidak efektif untuk menyingkap hana) atau aplikasi yang diuji p baik (terbebas dari failure). ini, langkah selanjutnya yang uk membuktikan kemungkinan cara menambahkan mutan ke

n eksperimen dan failure yang kasus tersebut berhasi tersingkap n yang terdapat pada studi kasus

unakan pada tulisan ini masih rena itu, penulis berharap agar anjutnya dapat membangkitkan ompleks sehingga failure yang banyak.

[9] "Software Testing". diakses pada tanggal 09 Januari 2012 dari http://en.wikipedia.org/wiki/Software_testing.

[10] Chen. Tsong Yueh. Ying Liu. dan Antony Tang, Metamophic Testing and Testing with Special Values, Swinburne University of Technology. Australia

[11] Zhou Zhi Quan. Tse T. H.. Kuo F. –C.. Chen T.Y., "Automated Functional Testing of Web Search Engine in the Absence of an Oracle". 2007.

[12] Rich. Hanna. Sumiati. "Penerapan Metamorphic Testing (MT) Studi Kasus: Fitur Penghitung Nilai Pada Sistem Informasi Akademik Politeknik Informatika Del dan Aplikasi Tebak Kendaraan". Laporan Tugas Askhir Politeknik Informatika Del. 2012

...ky, "An In-depth Study of Adaptive ...for Testing Program with Complex Input ...dissertation, Swinburne University of ...

...huang D. H., Tse. T. H., Yang Zongyuan, ...n Chen T. Y., " Metamorphic Testing and ..., 2004.

...ng, and S. Yiu. Metamorphic testing: a new ...erating text test cases. Technical Report ..., Department of Computer Science, Hong ... of Science and Technology. Hong Kong,

...nsky, T. Y. Chen, D. Grant, F.C. Kuo, and ...ing of heuristic methods: A case study of ...", in In Proceedings of the 3rd IFIP CEE ...oftware Engineering Techniques(CEE-SET

...) Program Testing Method, Prentice-Hall. ...EEE standard for software testing ...EEE Std 829-1983.