

# LIFECYCLE, ITERATION, AND PROCESS AUTOMATION WITH SMS GATEWAY

**Fenny**

Computer Engineering Department, Faculty of Engineering, Binus University  
Jl.K.H.Syahdan no 9, Palmerah, Jakarta Barat 11480  
fennysimar@gmail.com

## ABSTRACT

*Producing a better quality software system requires an understanding of the indicators of the software quality through defect detection, and automated testing. This paper aims to elevate the design and automated testing process in an engine water pump of a drinking water plant. This paper proposes how software developers can improve the maintainability and reliability of automated testing system and report the abnormal state when an error occurs on the machine. The method in this paper uses literature to explain best practices and case studies of a drinking water plant. Furthermore, this paper is expected to be able to provide insights into the efforts to better handle errors and perform automated testing and monitoring on a machine.*

**Keywords:** *lifecycle, iteration, automated testing, software engineering*

## ABSTRAK

*Menghasilkan system perangkat lunak berkualitas membutuhkan pemahaman pada indicator kualitas perangkat lunak, deteksi kecacatan, dan otomatisasi pengujian. Makalah ini bertujuan mengangkat desain dan proses pengujian otomatis pada mesin pompa air di suatu pabrik air minum. Makalah ini mengusulkan bagaimana pengembang perangkat lunak dapat melakukan maintainability dan meningkatkan keandalan system pengujian otomatis serta melaporkan keadaan abnormal bila terjadi error pada mesin. Metode pada makalah ini menggunakan studi literatur dengan memaparkan best practice dan studi kasus pabrik pemurnian air minum. Selanjutnya diharapkan paper ini dapat memberikan wawasan mengenai upaya lebih baik dalam menangani error dan melakukan pengujian dan monitoring otomatis pada suatu mesin.*

**Kata kunci:** *lifecycle, iteration, pengujian otomatis, rekayasa perangkat lunak*

## PENDAHULUAN

Pengembang perangkat lunak menghadapi tekanan untuk menghasilkan perangkat lunak berkualitas. Selain itu, mereka juga harus mampu merancang, mengimplementasikan, diuji, dan membuat sistem pengujian otomatis agar perangkat lunak mereka dapat bekerja dengan baik. Cara ini terbukti menjadi salah satu strategi yang paling ampuh untuk meningkatkan efisiensi pemeriksaan kualitas perangkat lunak (De Lemos, *et al.*, 2013). Cara ini menghemat hingga 50 persen daripada 80 pengujian perangkat lunak secara manual. Pengembang perangkat lunak dapat merancang dan mengembangkan praktis otomatis sistem pengujian perangkat mereka dengan metode menghilangkan redundansi perangkat lunak. Dengan demikian para pengembang dapat bekerja dan menguji kode-kode secara lebih efisien dan lebih efektif (Jalote, *et al.*, 2012). Otomatisasi pengujian juga memudahkan menanamkan *update/versi* baru ke dalam sistem perangkat lunak dan melakukan validasi otomatis untuk fitur baru ditambahkan. Ketika rilis terbaru diluncurkan, maka kegiatan instalasi berulang-ulang dapat digantikan dengan metode *iteration* untuk memvalidasi kebutuhan fungsional dan non-fungsional dari sistem tersebut. Pengujian otomatis tampaknya ideal untuk proses ini, seperti, dengan desain, sistem komputasi melakukan sejumlah besar tindakan berulang.

Metode ini juga penting untuk membantu pengembang perangkat lunak dalam penerapan pengujian otomatis pada sistem operasi tertentu (Völter, *et al.*, 2013). Dalam melakukan hal ini perlu memahami proses perancangan pada *database* dan *system* informasi yang dipakai. Kemudian, dalam makalah ini akan dipakai pengujian otomatis untuk menentukan kualitas aplikasi SMS dan melihat penerapan pengujian otomatis subjek kita bekerja Desktop Assistant, aplikasi perangkat lunak SMS.

Selanjutnya pertanyaan penting dalam makalah ini yaitu: (1) Bagaimana cara meningkatkan kualitas pengujian otomatis pada aplikasi dalam mengupdate data dan memberikan informasi mengenai keadaan mesin melalui sms. (2) Bagaimana kualitas aplikasi dapat merespon permintaan informasi yang dibutuhkan oleh user. (3) Bagaimana aplikasi dapat menjalankan fungsi-fungsi sesuai dengan design.

Dari penerapan pengujian otomatis itu, akan menghasilkan fitur SMS sebagai bukti bahwa aplikasi pengujian otomatis itu telah bekerja dengan baik (Butt, *et al.*, 2012). Yang memfasilitasi deteksi kerusakan mesin pabrik. Pengujian Aplikasi SMS Gateway bertujuan mengangkat desain dan proses pengujian otomatis pada mesin pompa air di suatu pabrik air minum. Melalui *interface*-nya pengguna dapat mengelola kontak, pesan instan, dan status pengiriman. Kami menyelidiki dan menganalisa berbagai metodologi pengujian otomatis dan alat dan akhirnya memantau dan melaporkan bila terjadi keadaan abnormal pada perangkat yang tersebut sebagai komponen inti dari kerangka pengujian otomatis ini. Dengan pengujian aplikasi SMS Gateway diharapkan para pengembang perangkat lunak dapat melakukan maintainability dan meningkatkan keandalan system pengujian otomatis serta melaporkan keadaan abnormal bila terjadi *error* pada mesin.

Paper ini terdiri dari lima bagian. Bagian 1 berisi latar belakang masalah. Bagian 2 berisi gambaran metode pengujian otomatis yang ada. Bagian 3 berisi persyaratan dan kerangka pengujian otomatis termasuk persyaratan kasus dan setup prasyarat. Bagian 4 berisi desain dan implementasi sistem pengujian otomatis. Bagian 5 berisi hasil evaluasi dan kesimpulan.

### Definisi Pengujian Otomatis

Pengujian otomatis adalah penggunaan perangkat lunak untuk mengontrol pelaksanaan tes, perbandingan hasil aktual dengan hasil prediksi, pengaturan dari prasyarat pengujian, dan kontrol pengujian dan pelaporan fungsi lainnya (Robertson & Robertson, 2012). Pengujian otomatis memungkinkan komputer mendeteksi dan melaporkan kesalahan dan masalah rutin secara otomatis tanpa interaksi manusia. Dengan cara ini, sistem komputasi dapat melakukan kegiatan rutindan

melaporkan pengujian tersebut sehingga secara signifikan mengurangi upaya pengujian manual. Selain melaporkan kesalahan dan masalah, pengujian otomatis dapat memenuhi tujuan-tujuan lain (Raneburger, *et al.*, 2014). Misalnya, ketika alat pengujian otomatis menghasilkan program untuk mengeksekusi pengujian rutin, maka perlu menanamkan *tool* tertentu yang sesuai persyaratan spesifikasi.

Untuk menguji sebuah perangkat lunak, maka perlu menganalisis perangkat lunak tersebut dan menyelidiki fungsi melaluikusus (Naumann, *et al.*, 2015). Bila terjadi masalah maka perlu menyelidiki situasi dan mengambil tindakan pemeriksaan sebagai fungsi otomatis dari perangkat lunak tersebut. Jika *output* seperti yang diharapkan, lulus ujian (Zugal, *et al.*, 2012). Jika tidak, maka gagal uji. Karena tidak semua kasus mengizinkan otomatisasi pengujian, *tester* perlu memilih kasus dengan akurat (O'Leary, *et al.*, 2012). Hasilnya kemudian dimunculkan melalui aplikasi SMS gateway. Dengan fitur ini, maka dapat dilakukan fungsi otomatis melalui pengiriman SMS, pesan instan, dan manajemen kontak, yang memungkinkan melalui antarmuka pengguna (Amalfitano, *et al.*, 2012). Dengan memantau *user interface*, maka *computer*/HP dapat memvalidasi fungsinya. Sebagai contoh, pada penerimaan pengiriman SMS, pengiriman *popup window* masuk muncul. Identifikasi pengirim dan nama akan muncul di jendela. Dengan memvalidasi keberadaan jendela ini dan identifikasi pengirim dan nama, seseorang dapat lulus atau gagal ujian menerima pengiriman masuk. Oleh karena itu, sebagian besar pengujian bersifat pengujian antarmuka pengguna grafis (GUI) (Christie, 2012). Alat uji ini diterapkan untuk pemantauan mesin pompa air. Dengan cara ini dapat diketahui silinder mesin pompa yang berjalan dengan baik ataupun mengalami kegagalan.

## Life Cycle Perangkat Lunak

Life Cycle atau siklus hidup adalah urutan fase di mana proyek menentukan, desain, mengimplementasikan, tes, dan mempertahankan sebuah perangkat lunak untuk pengujian perangkat tertentu (Andersson, *et al.*, 2013). Teknik ini memudahkan pengembang untuk menentukan spesifikasi sesuai dengan kebutuhan pengguna adalah tujuan rekayasa perangkat lunak. Dalam hal ini, struktur program perlu disesuaikan agar memenuhi spesifikasi (Bashir, 2012). Perlu memahami dinamika perubahan kebutuhan pengguna. Selanjutnya tugas perlu dibagikan kepada anggota tim. Secara umum, Rekayasa Perangkat Lunak didefinisikan sebagai seni mengembangkan perangkat lunak berkualitas tepat waktu dan sesuai anggaran sebagai Trade-off antara kesempurnaan dan kendala yang dihadapi. Dengan demikian, Rekayasa Perangkat Lunak merupakan aktivitas pengembangan perangkat lunak dan mengatasi masalah berulang (*iterative*) yang mengikuti model dasar. Selama ini, masalah umum dihadapi pengembang perangkat lunak ialah sering menghadapi kendala dari keinginan pelanggan untuk menyatakan semua persyaratan eksplisit (Tsui, 2013). Selain itu, terdapat kesulitan mengakomodasi ketidakpastian alamiah yang ada pada awal banyak proyek dimana pengembang harus menciptakan algoritma yang dapat dikembangkan lebih lanjut sesuai perkembangan dan kebutuhan lebih lanjut dari pelanggan mereka. Ketidakmampuan mencapai tuntutan pelanggan sampai akhir waktu proyek dapat menjadi kesalahan besar.

## Pengembangan Iteratif

Pengembangan iteratif bertujuan mendapatkan yang benar pertama kalinya, sehingga mengintegrasikan, memvalidasi dan menguji program sesering mungkin (Poppendieck & Cusumano, 2012). Pengembangan berulang meningkatkan kualitas perangkat lunak agar dapat mengintegrasikan fungsi baru ke dalam model lama. Selanjutnya, dalam proses analisis kebutuhan, pengembang perlu menentukan apa yang harus dilakukan sistem agar dapat bekerja sesuai spesifikasi (Jalote, *et al.*, 2012).

## Menentukan Desain

Desain adalah proses menentukan bagaimana sistem dengan perilaku tertentu akan terwujud menjadi komponen perangkat lunak (Völter, *et al.*, 2013). Hasil dari desain perangkat lunak ialah

arsitektur dan dokumen desain rinci. Desain berorientasi objek memberikan model yang menggambarkan operasi sistem diimplementasikan dengan objek yang saling berinteraksi terutama melalui kelas-kelas yang merujuk satu sama lain dan berhubungan dengan atribut dan operasi (Butt, *et al.*, 2012). Hukum Conway menyatakan bahwa pengembang perangkat lunak perlu memahami pengelolaan *system* desain untuk menghasilkan desain dengan struktur yang tepat yang dapat memudahkan dalam pengembangan, modifikasi, dan integrasi produk baru ke dalam system keseluruhan (Robertson & Robertson, 2012). Untuk itu perlu dilakukan analisis berorientasi objek dalam model sistem menggambarkan kelas-kelas objek yang ada dalam system, bagaimana tanggung jawab kelas-kelas dan hubungan antara kelas-kelas. Dalam beberapa kasus, penggunaan dan skenario menggambarkan operasi yang dapat dilakukan pada system dan operasi. Setelah desain dapat ditentukan, maka selanjutnya ialah pelaksanaan dan pengujian kesalahan (*error*). Dalam hal ini, tahap pelaksanaan merupakan kegiatan mengembangkan desain menjadi solusi perangkat lunak jadi sesuai kebutuhan pelanggan (Ranenburg, *et al.*, 2014). Pengujian perlu dilakukan untuk memvalidasi bahwa perangkat lunak itu telah memenuhi kebutuhan pelanggan.

## **Sistem Pengolahan Informasi**

Sistem pada dasarnya adalah suatu kumpulan atau himpunan dari unsur, komponen atau variable-variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu (Naumann, *et al.*, 2015). Sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu memiliki komponen saling berhubungan, memiliki batasan dan lingkungan internal dan eksternal (O'Leary, *et al.*, 2012). *System* yang baik memiliki penghubung *system* dan masukan/keluaran yang kemudian diolah oleh pengolah *system* menghasilkan target *output*/keluaran. System juga mempunyai beberapa komponen, diantaranya: Perangkat keras, Perangkat lunak (Sistem operasi, Database, Program aplikasi), Data, dan Prosedur (Amalfitano, *et al.*, 2012).

## **Short Message Service (SMS)**

Short Message Service atau yang lebih dikenal dengan istilah SMS berfungsi untuk memberikan layanan pengiriman pesan teks singkat antar perangkat telepon genggam/ telepon bergerak yang memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric*. Keuntungan yang diberikan oleh layanan SMS antara lain murah, praktis, dan bebas tempat (Christie, 2012).

## **Usabilitas pengujian**

Untuk mengetahui seberapa tinggi manfaat suatu alat pengujian perangkat lunak, maka digunakan konsep usabilitas. Usabilitas dianggap penting sebagai pedoman dalam mengetahui apakah perangkat lunak tertentu dapat diterapkan dalam berbagai setting yang berbeda seperti pada *desktop* GUI dan *smartphone* atau tampilan lainnya. Acuan ini sesuai dengan pendapat Amalfitano, *et al.* (2012). Dengan melihat konsep Usabilitas maka dapat ditentukan seberapa jauh suatu perangkat lunak yang diuji itu memiliki kompatibilitas untuk bekerja dengan berbagai komponen perangkat keras berbeda misalnya menggunakan HP ataupun *computer/website*.

## **METODE**

Metode penelitian ini menggunakan studi literatur untuk mendapatkan teori-teori yang berhubungan dan mendukung penulisan pada artikel ini. Bahan studi pustaka didapat dari jurnal-jurnal, artikel, buku, maupun referensi lainnya yang membantu dalam proses pengolahan dan analisa data. Selain itu, dilakukan studi pustaka pada penelitian sebelumnya untuk mendapatkan informasi yang dibutuhkan.

Selain studi literatur, kunjungan pada objek dan melakukan wawancara pada 3 bagian di suatu pabrik pemurnian air di PT ABC di Jakarta. Metode ini bertujuan mengamati suatu keadaan akhir dan kemudian ditarik mundur untuk mengatasi terulangnya suatu kejadian atau kekeliruan dilakukan perangkat lunak. Proses ini kemudian dipakai dan dibuat *coding* sebagai fungsi rutin untuk pengujian otomatis kualitas perangkat lunak. Dalam melakukan analisis sistem perangkat lunak, perlu memahami proses perancangan pada database dan system informasi yang dipakai. Kemudian, dalam makalah ini akan dipakai pengujian otomatis untuk menentukan kualitas aplikasi SMS dan melihat penerapakan pengujian otomatis subjek kita bekerja Desktop Assistant, aplikasi perangkat lunak SMS.

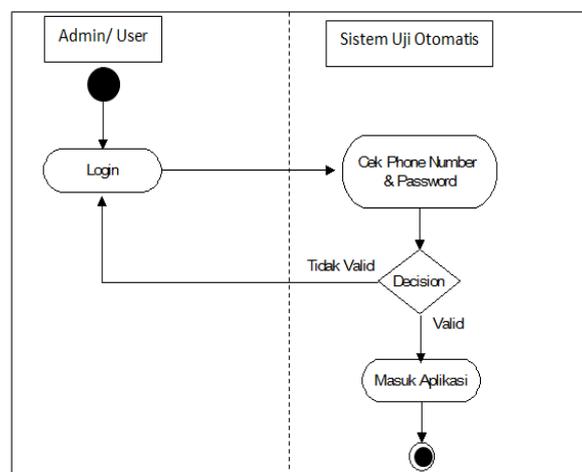
Untuk menguji sebuah perangkat lunak untuk pengujian mesin pompa air (MPA), maka perlu menganalisis perangkat lunak dan menyelidiki fungsi melalui kasus. Sebuah kasus memerlukan penyelidikan dari set tertentu dari tindakan yang dilakukan oleh fungsi tertentu dari perangkat lunak dan *output* tindakan pemeriksaan. Jika output seperti yang diharapkan, lulus ujian. Jika tidak, maka gagal uji. Karena tidak semua kasus mengizinkan otomatisasi pengujian, *tester* perlu memilih kasus dengan akurat. Hasilnya kemudian dimunculkan melalui aplikasi SMS. Fitur Desktop Asisten mengelola pengiriman SMS, pesan instan, dan manajemen kontak, yang memungkinkan melalui antarmuka pengguna. Dengan memantau *user interface*, satu dapat memvalidasi fungsinya.

## Analisis dan Perancangan

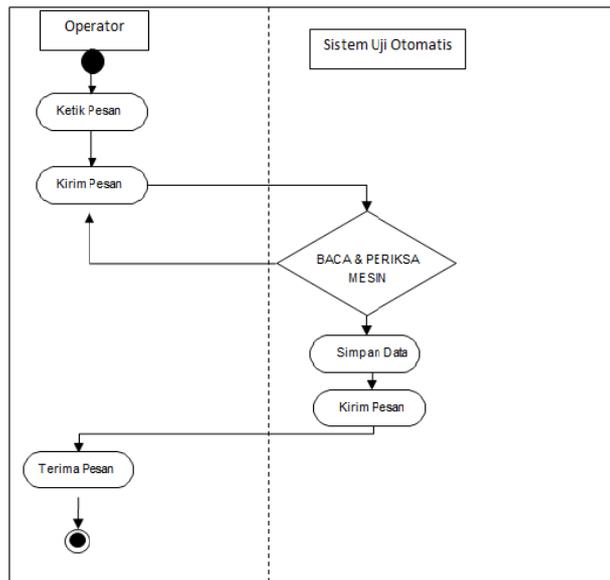
Analisis masalah penting untuk mengatasi perbedaan kenyataan antara desain dan kondisi di lapangan. Akurasi data antara laporan dengan kenyataan di lapangan (Andersson, *et al.*, 2013). Oleh karena itu aplikasi pengujian otomatis pompa air ini memudahkan pemantauan konsumsi air bersih pada pabrik pemurnian air di PT ABC di Jakarta. Saat ini PT ABC tidak memiliki alat uji otomatis untuk pemantauan dan perhitungan konsumsi air bersih sehingga banyak terjadi masalah mengenai pemakaian air bersih. Dengan demikian, perusahaan dapat bersiap bila terjadi kekurangan air bersih di pabrik mereka.

## Perancangan Sistem

Aplikasi ini berbasis PHP dengan menggunakan system database Mysql dan *software* sms gateway sebagai perantara untuk sarana penerima SMS dan mengirimkan SMS respon secara otomatis (Bashir, 2012). Selanjutnya, aplikasi ini akan secara otomatis mengirimkan SMS dengan format *autoresponse*. Pada aplikasi yang akan dibangun, proses aplikasi dapat digambarkan diagram aktifitas seperti pada gambar 1 sedangkan pada aplikasi yang akan dibangun, proses aplikasi dapat digambarkan diagram aktivitas seperti pada gambar 2.



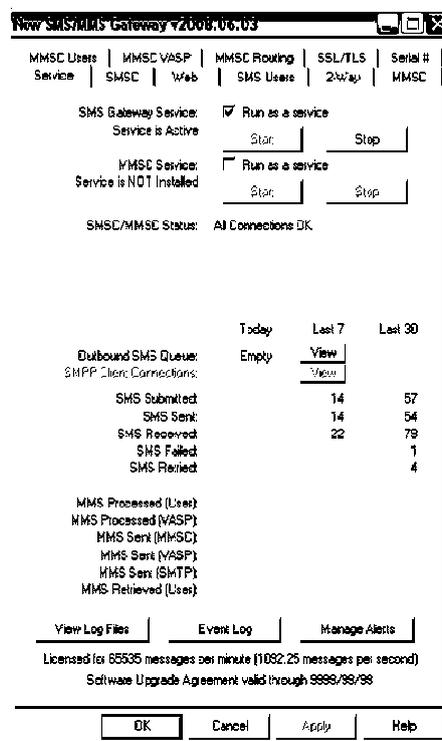
Gambar 1 Diagram Aktifitas Login



Gambar 2 Diagram Aktifitas Kirim dan Terima SMS

## Rancangan SMS Center

SMS Center / sms gateway yang dapat menerima sms dan mengirimkan sms secara otomatis (auto respon) sesuai dengan *keyword* yang dikirimkan oleh operator (Tsui, 2013). Dalam aplikasi ini penulis menggunakan NOW SMS sebagai SMS Center/ SMS Gateway.



Gambar 3 Perancangan SMS Gateway dengan NOW SMS

## HASIL DAN PEMBAHASAN

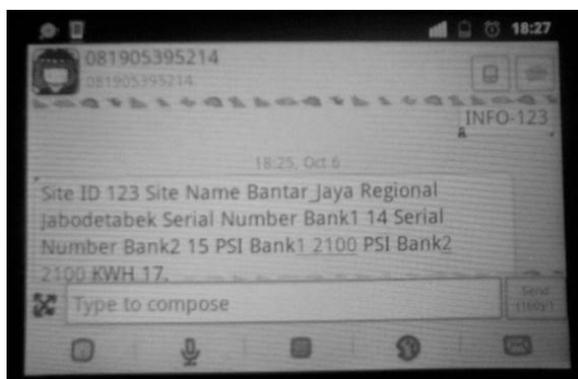
Dari pengujian yang telah dilakukan dan diperoleh *output* dari setiap inputan, dari hasil tersebut penulis merangkum analisa hasil pengujiannya. Pertama, pengujian login, jika user memasukan nomor telepon, *password* dan *captcha* sesuai dengan data yang ada didalam *database* maka aplikasi akan meneruskan ke halaman HOME sesuai dengan tingkatan atau level *user* baik sebagai *user* maupun admin. Dan jika *user* memasukan nomor telepon, *password* dan *captcha* tidak sesuai dengan data yang ada dalam *database* maka aplikasi akan kembali ke halaman LOGIN. Kedua, tambah *user*, hanya *user* berlevel admin yang dapat menambahkan *data user* dalam *data base*. Jika *user* memasukan semua data inputan (nomor telepon, nama lengkap, *password*, level dan regional) dengan lengkap dan nomor telepon belum ada didalam *database* maka aplikasi akan menyimpan data inputan ke dalam *database* dan aplikasi akan ke halaman LIHAT USER. Dan jika data inputan ada yang tidak terisi atau nomor telepon telah tersimpan dalam *database* maka aplikasi akan kembali ke halaman TAMBAH USER.

Ketiga, *input* mesin pompa air *on site*, hanya *user* berlevel admin yang dapat memasukan inputan mesin pompa air *on site* dalam aplikasi. *User* harus terlebih dahulu memasukan Site ID Pompa Air dan jika Site ID salah (tidak ada didalam *database*) ataupun telah ada data mesin pompa air *on site* maka aplikasi akan kembali ke Inputan Site ID namun jika data benar maka aplikasi akan ke halaman inputan mesin pompa air *on site*. Dalam halaman inputan mesin pompa air *on site* *user* harus memasukan inputan (Serial No Tabung1, Serial No Tabung2, PSI Tabung 1, PSI Tabung 2, KWH dan Konsumsi). Keempat, SMS “INFO”. Aplikasi akan mengirimkan data mengenai Nama Site, Regional, Serial Number Tabung1, Serial Number Tabung2, PSI Tabung1, PSI Tabung2 dan KWH kepada operator dengan format INFO-SITE ID.

Kelima, SMS “REF1”. Operator mengirimkan SMS dengan format yang ditentukan dan aplikasi akan mengirimkan notifikasi Refiling Air Bersih OK dan mengenai status pemakain air bersih pabrik, jika operator mengirimkan SMS dengan format yang salah maka aplikasi akan mengirimkan notifikasi hasil uji otomatis bila terjadi *error* pada mesin. Keenam, SMS “CONS”. Aplikasi akan mengirimkan data konsumsi air bersih pabrik dan kemungkinan terjadinya kebocoran jika operator mengirimkan SMS dengan format yang telah ditentukan, jika format salah maka aplikasi akan mengirimkan SMS notifikasi kesalahan.

### Tampilan Hasil Uji Otomatis

Tampilan hasil uji otomatis merupakan respon yang mengirimkan informasi mengenai data-data mengenai mesin pompa air bersih di PT ABC.



Gambar 4 Tampilan Hasil Uji Otomatis

```

if($pisah[0]=="INFO")
{
//Periksa Site ID
$info=mysql_fetch_array(mysql_query ("select *from pabrik ABC where
site_id='$pisah[1]'"));
$info1=mysql_fetch_array(mysql_query ("select *from cyl_on_site where
site_id='$pisah[1]'"));
if($pisah[1]==$info['site_id'])
//Pengiriman Pesan (Jika Site ID Tersedia)
{header("Location:http://127.0.0.1:8800/?PhoneNumber=$sender&Text=Site+ID+$pisah
ah[1]+Site+Name+$info[site_name]+Regional+$info[nama_reg]+Serial+Number+Tab
ung1+$info1[serial_no_tabung1]+Serial+Number+Tabung2+$info1[serial_no_tabung
2]+PSI+Tabung1+$info1[psi_tabung1]+PSI+Tabung2+$info1[psi_tabung2]+KWH+$i
nfo1[kwh].");}

Else
//Pengiriman Pesan (Jika Site ID Tidak Tersedia)

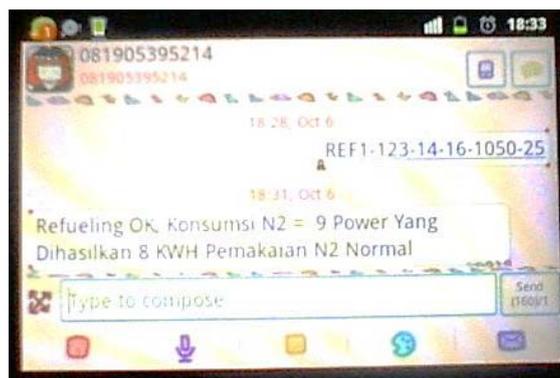
{header("Location:http://127.0.0.1:8800/?PhoneNumber=$sender&Text=Alamat+Site
+ID+$pisah[1]+tidak+tersedia+periksa+kembali+site+ID.");}
}

```

Gambar 5 Coding Tampilan Hasil Uji Otomatis

## Tampilan bila hasil uji otomatis berjalan baik

Tampilan bila hasil uji otomatis berjalan baik merupakan respon yang dikirimkan oleh aplikasi untuk mengetahui apakah pada tabung air bersih terjadi masalah atau tidak. Bila tidak terjadi masalah maka akan muncul SMS konfirmasi OK dan normal.



Gambar 6 Tampilan Respon  
Bila Hasil Uji Otomatis Berjalan Baik

```

//Pengiriman Pesan (Refiling Air Bersih OK) & Update Database
{header("Location:http://127.0.0.1:8800/?PhoneNumber=$sender&Text=Refiling Air
Bersih+OK,+Konsumsi+N2+===+$cons+Power+Yang+Dihasilkan+$power+KWH+$r
emark");
mysql_query("update cyl_on_site set serial_no_tabung1 ='$ref[serial_no_tabung2]',
psi_tabung1
='$pisah[6]',serial_no_tabung2='$cek4',
psi_tabung2='2100',kwh='$pisah[7]', konsumsi='$cons' where site_id='$pisah[1]'");
mysql_query("update cylinder set status='kosong' where serial_no='$pisah[2]'");
mysql_query("update cylinder set status='kosong' where serial_no='$pisah[3]'");
mysql_query("update cylinder set status='$conf[site_name]' where
serial_no='$pisah[4]'");
mysql_query("update cylinder set status='$conf[site_name]' where
serial_no='$pisah[5]'"); }

//Pengiriman Pesan bila terjadi error pada mesin pompa Air
else

{header("Location:http://127.0.0.1:8800/?PhoneNumber=$sender&Text=Cek+Kembal
i+Serial+Number+Cylinder");} }
else
{header("Location:http://127.0.0.1:8800/?PhoneNumber=$sender&Text=Cek+Kembal
i+Serial+Number+Cylinder(Serial+Tidak+Cocok");} }

if ($conf['configuration']=="3+3")
//Perhitungan Konsumsi & Remark
{$cons=($ref['psi_tabung1']+$ref['psi_tabung2']-$pisah[8])*3/350;
$power=$pisah[9]-$ref['kwh'];
if($power==0){$remark ="KWH NOL, Cek Konsumsi N2 dan Selang";}
else {$cons1=$cons/$power;
if($cons1>=1.2){$remark="Pemakaian+N2+Berlebihan,+Cek+Selang";}
else {$remark="Pemakaian+N2+Normal";}
}
}

```

Gambar 7 Coding Tampilan Pesan Bila Terjadi Error pada Perangkat Lunak atau pada Mesin Pompa Air

## SIMPULAN

Kualitas Perangkat Lunak dapat dilihat dari seberapa baik perangkat lunak itu berjalan sesuai dengan persyaratan desain yang ditetapkan. Dalam penelitian ini, perangkat lunak untuk pengujian mesin pompa air memenuhi kriteria seperti kegunaan, usabilitas, diperpanjang, kompatibilitas, fungsionalitas, ketepatan, ketahanan, efisiensi, dan ketepatan waktu seperti disarankan oleh Jalote (2012). *Usability* mengacu pada kemudahan dengan yang satu dapat menggunakan perangkat lunak (Poppendieck & Cusumano, 2012). Jika perangkat lunak sangat bisa digunakan, orang-orang dengan latar belakang yang berbeda dan dengan berbagai kualifikasi dapat belajar untuk menggunakannya dengan mudah (Völter *et al.*, 2013).

Dalam penelitian ini, perangkat lunak telah diuji untuk menentukan tingkat usabilitas. Dikarenakan usabilitas adalah faktor kualitas perangkat lunak, maka digunakan acuan kemampuan elemen perangkat lunak untuk bekerja dalam aplikasi yang berbeda, yang dalam penelitian ini bekerja dengan menggunakan SMS gateway maupun komputer (Amalfitano, *et al.*, 2012). Dengan cara

ini, kualitas dan kinerja perangkat lunak memiliki kompatibilitas untuk bekerja dengan berbagai komponen perangkat keras misalnya menggunakan HP ataupun *computer/website*. Selain kriteria kualitas perangkat lunak ini, perangkat lunak harus benar, kuat, efisien, dan tepat waktu (Andersson, *et al.*, 2013). Tindakan kebenaran bagaimana produk perangkat lunak untuk uji otomatis dalam penelitian ini juga mampu melakukan tugas-tugas yang telah ditentukan. Dengan kata lain, alat uji otomatis perangkat lunak ini mampu melakukan perilaku yang diharapkan serta melaporkan situasi abnormal. Kekurangan dari penelitian ini ialah tidak diperluas pada usability dalam perangkat komputer *desktop*.

## DAFTAR PUSTAKA

- Amalfitano, D., Fasolino, A. R., Tramontana, P., De Carmine, S., & Memon, A. M. (2012, September). Using GUI ripping for automated testing of Android applications. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (pp. 258-261). ACM.
- Andersson, J., Baresi, L., Bencomo, N., de Lemos, R., Gorla, A., Inverardi, P., & Vogel, T. (2013). Software engineering processes for self-adaptive systems. In *Software Engineering for Self-Adaptive Systems II* (pp. 51-75). Springer Berlin Heidelberg.
- Bashir, I., & Goel, A. L. (2012). *Testing object-oriented software: life cycle Solutions*. Springer Science & Business Media.
- Butt, S. M., Ahmad, W. F. W., & Fatimah, W. (2012). An Overview of Software Models with Regard to the Users Involvement. *International Journal of Computer Science Issues (IJCSI)*, 9(3), 1.
- Christie, A. M. (2012). *Software process automation: the technology and its adoption*. Springer Science & Business Media.
- De Lemos, R., Giese, H., Müller, H. A., Shaw, M., Andersson, J., Litoiu, M., & Wuttke, J. (2013). Software engineering for self-adaptive systems: a second research roadmap. *Software Engineering for Self-Adaptive Systems II* (pp. 1-32). International Seminar, Dagstuhl Castle, Germany, October 24-29.
- Jalote, P. (2012). *An integrated approach to software engineering*. Springer Science & Business Media.
- Naumann, S., Kern, E., Dick, M., & Johann, T. (2015). Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects. In *ICT Innovations for Sustainability* (pp. 191-205). Springer International Publishing.
- O'Leary, P., McCaffery, F., Thiel, S., & Richardson, I. (2012). An agile process model for product derivation in software product line engineering. *Journal of Software: Evolution and Process*, 24(5), 561-571.
- Poppendieck, M., & Cusumano, M. (2012). Lean software development: A tutorial. *Software, IEEE*, 29(5), 26-32.
- Raneburger, D., Popp, R., Kaindl, H., Armbruster, A., & Šajatović, V. (2014). An Iterative and Incremental Process for Interaction Design through Automated GUI Generation. In *Human-Computer Interaction. Theories, Methods, and Tools* (pp. 373-384). Springer International Publishing.

- Robertson, S., & Robertson, J. (2012). *Mastering the requirements process: Getting requirements right*. Addison-wesley.
- Tsui, F. F., Karam, O., & Bernal, B. (2013). *Essentials of software engineering*. Jones & Bartlett Publishers.
- Völter, M., Stahl, T., Bettin, J., Haase, A., & Helsen, S. (2013). *Model-driven software development: technology, engineering, management*. John Wiley & Sons.
- Zugal, S., Pinggera, J., & Weber, B. (2012). Toward enhanced life - cycle support for declarative processes. *Journal of Software: Evolution and Process*, 24(3), 285-302.