

PENGUJIAN *WHITE BOX* PADA APLIKASI *DAILY ACTIVITY MONITORING* KARYAWAN BERBASIS WEB

Risah Subariah

Universitas Pamulang, Jl. Raya Puspitek No.11, Serpong, Tangerang Selatan

E-mail: risah.subariah@gmail.com

ABSTRAK

PENGUJIAN *WHITE BOX* PADA APLIKASI *DAILY ACTIVITY MONITORING* KARYAWAN BERBASIS WEB. Aplikasi *daily activity monitoring* ini bertujuan untuk memantau aktivitas pekerjaan karyawan dan disesuaikan dengan target yang akan dicapai. Pengujian *white box* ini dilakukan untuk mengetahui aplikasi *daily activity monitoring* berjalan terstruktur dan juga memberikan informasi terkait aplikasi yang mungkin terdapat *source code* yang memiliki tingkat kerumitan yang tinggi pada programnya serta mengetahui kesalahan-kesalahan yang terjadi pada program yang dibuat. Hasil dari pengujian *white box* pada aplikasi *daily activity monitoring* dengan teknik *basis path testing* ini adalah tidak ditemukan galat dan semua pengujian berjalan dengan baik.

Kata kunci: pengujian *white box*, aplikasi, *daily activity*

ABSTRACT

WHITE BOX TESTING ON WEB-BASED EMPLOYEE ACTIVITY MONITORING APPLICATION. This *daily activity monitoring* application aims to monitor employee work activities and adjust it to the targets to be achieved. This *white box* test is carried out to find out the *daily activity monitoring* application is running in a structured manner and also provides information related to applications that may have source code that has a high level of complexity in the program and find out errors that occur in the program being made. The results of the *white box* testing on the *daily activity monitoring* application with this *path-based testing* technique are that no errors were found and all tests went well.

Keywords: *white box testing*, application, *daily activity*

1. PENDAHULUAN

Daily activity monitoring atau *monitoring* kegiatan harian yang dipantau oleh atasan serta dibandingkan dengan target yang telah ditentukan[1]. Dengan adanya aplikasi *daily activity monitoring manager* dapat mengolah data karyawan dan memantau kinerja karyawan setiap harinya dan dibandingkan dengan target yang telah direncanakan sebelum pekerjaan dimulai. Aplikasi *daily activity monitoring* ini sangat bermanfaat karena dapat diakses oleh setiap

karyawan yang ingin mengisi aktivitas hariannya. *Manager* karyawan dapat dengan mudah melihat aktivitas karyawan yang dilakukan setiap hari dan disesuaikan dengan rencana target yang dibutuhkan. Bila terdapat ketidaksesuaian antara aktivitas yang diisi dengan hasil yang dikerjakan, maka *manager* dapat dengan mudah melakukan penolakan terhadap aktivitas yang telah diisi karyawan pada aplikasi *daily activity monitoring*. Agar aplikasi dapat berjalan dengan baik, dibutuhkan sebuah pengujian *white box* yang bertujuan untuk mengetahui kelemahan dan juga kekurangan sistem yang telah dibuat. Sehingga dapat menjadi pertimbangan untuk dilakukan perbaikan sistem menjadi lebih baik dari sebelumnya. Evaluasi kinerja karyawan dapat dijadikan acuan penilaian kinerja karyawan sehingga lebih akurat dengan sistem[2].

Pengujian *white box* atau yang memiliki sebutan pengujian kotak putih adalah sebuah filosofi perencanaan *test case* atau uji kasus dengan menggunakan struktur kontrol yang dijelaskan kedalam bagian perancangan komponen yang *test case*. Pengujian dengan kotak putih dapat menghasilkan jalur *independent* pada modul telah dieksekusi minimal satu kali, melakukan struktur data internal untuk mengetahui kebenarannya, melakukan putaran pada batasan pada batas operasional, melakukan keputusan logis pada sisi benar dan salah[3].

Pengujian *white box* menggunakan teknik *basis path* yang dapat digunakan untuk mengukur kompleksitas logis desain prosedur untuk dijadikan pedoman dalam konsistensi jalur aplikasi, pengujian *statement* program minimal satu kali selama pengujian dan menghitung *cyclomatic complexity* untuk menentukan jumlah *independent path* untuk pengujian[4].

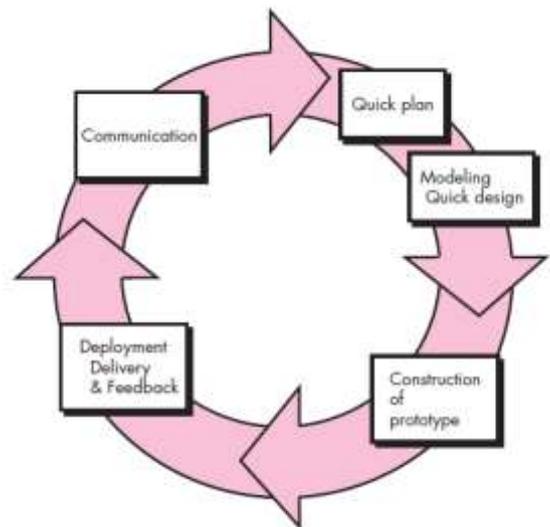
Berdasarkan latar belakang tersebut maka diperlukan sebuah pengujian *white box* pada

aplikasi *daily activity monitoring*. Tujuan penelitian ini diharapkan dapat mengetahui aplikasi *daily activity monitoring* berjalan terstruktur dan juga memberikan informasi terkait aplikasi yang mungkin terdapat *source code* yang memiliki tingkat kerumitan yang tinggi pada programnya serta mengetahui kesalahan-kesalahan yang terjadi pada program yang dibuat.

2. METODE

2.1 Model Prototype

Metode yang digunakan pada perancangan aplikasi *Daily Activity Monitoring* adalah metode dengan model *prototype*. *Prototype* adalah mekanisme dalam mengidentifikasi kebutuhan dalam sebuah perangkat lunak secara rinci dan detail mulai dari fungsi dan juga fitur yang ditawarkan dalam perangkat lunak.



Gambar 1. Ilustrasi Model Prototype (Sumber: Pressman, 2010)

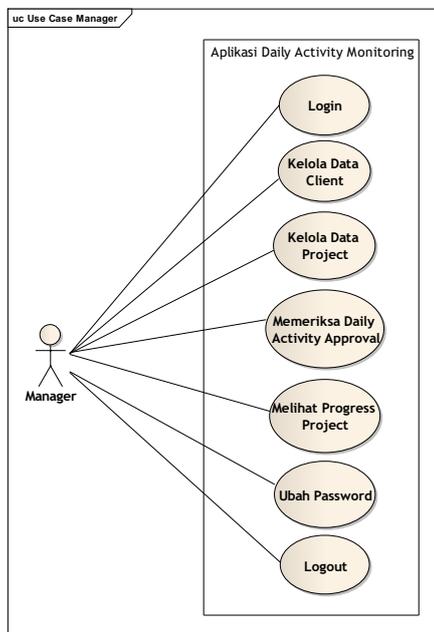
Tahapan yang ada dalam *prototype* adalah sebagai berikut:

- 1) Menganalisa kebutuhan pengguna, komunikasi dan pengumpulan data.
- 2) Membuat *design* sementara dengan cepat yang nantinya akan dikembangkan kembali (*quick design*).
- 3) Membuat sebuah *prototype* perangkat, pengujian dan penyempurnaan termasuk kedalamnya.
- 4) Mengevaluasi *prototype* dan menganalisis kebutuhan penggunaanya.

- 5) Memperbaiki *prototype*, membuat tipe yang sesungguhnya dari penyempurnaan *prototype*.
- 6) Produksi akhir, perangkat diproduksi dengan hasil yang sesuai dan dapat digunakan oleh pengguna.

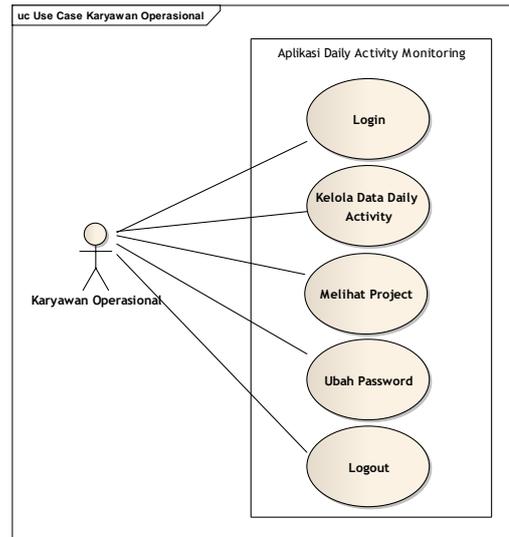
2.2 Diagram Sistem

Analisa pada sistem berjalan diperlukan dan dideskripsikan kedalam diagram[5]. Analisa data digunakan untuk mengoptimalkan data yang telah ada[6]. Aplikasi *daily activity monitoring* ini digambarkan pada diagram-diagram dibawah mulai dari proses pengisian *daily activity* hingga proses approval dari *manager* pada saat mengecek aktivitas yang telah dimasukkan oleh karyawan:



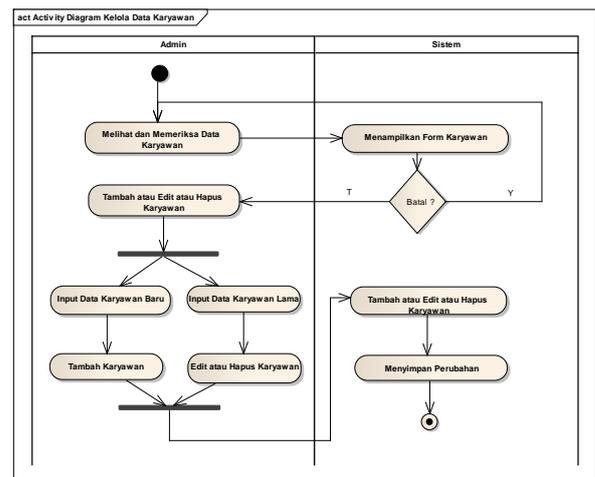
Gambar 2. Use Case Manager

Use case diagram yang dibuat untuk menggambarkan setiap tugas para actor pada sistem[7]. Diagram diatas menjelaskan alur atau proses dari *manager* mulai dari mengelola data *client*, data *project*, memeriksa *daily activity approval* serta mengecek *progress project* pada aplikasi *daily activity monitoring*.



Gambar 3. Use Case Karyawan Operasional

Diagram diatas menjelaskan alur atau proses dari karyawan operasional mulai dari mengelola *daily activity*, melihat *project*.

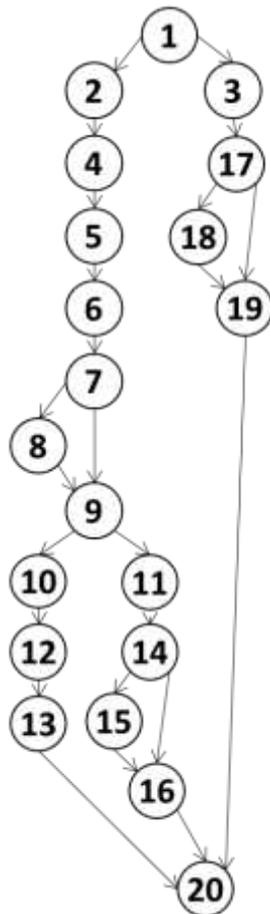


Gambar 4. Activity Diagram Kelola Data Karyawan

Pada diagram diatas menjelaskan bahwa *admin* melihat dan memeriksa data, sistem akan menampilkan *form* karyawan, jika proses batal maka sistem akan kembali ke proses awal, jika proses berhasil maka *admin* dapat menambah, merubah atau menghapus data karyawan. Jika *admin* telah melakukan penambahan atau perubahan pada data karyawan, maka sistem akan menyimpan perubahan.

b. *Flowgraph* kelola data karyawan

Dibawah ini adalah *flowgraph* yang dihasilkan dari pemetaan *source code* kelola data karyawan pada gambar 9:



Gambar 10. *Flowgraph* Kelola Data Karyawan

c. *Cyclomatic Complexity*

Berdasarkan *flowgraph* kelola data karyawan maka kita akan menghitung *cyclomatic complexity* yang dapat menentukan banyaknya *independent path* yang merupakan *basis path* seperti dibawah ini:

Rumus:

$$V(G) = E - N + 2$$

$$V(G) = 24 - 20 + 2$$

$$= 6$$

$$V(G) = P + 1$$

$$V(G) = 5 + 1$$

$$= 6$$

Keterangan:

E : Edge

N : Node

P : Predicate node

d. *Independent Path*

Hasil dari perhitungan *cyclomatic complexity* menghasilkan 6 path yaitu:

Path 1: 1-2-4-5-6-7-8-9-10-12-13-20

Path 2: 1-2-4-5-6-7-9-10-12-13-20

Path 3: 1-2-4-5-6-7-8-9-11-14-15-16-20

Path 4: 1-2-4-5-6-7-8-9-11-14-16-20

Path 5: 1-3-17-18-19-20

Path 6: 1-3-17-19-20

3.3 Analisa dan Hasil Uji

Pengujian diawali dengan pemetaan *source code*, membuat *flowgraph*, dan menghitung *Cyclomatic Complexity*. Kemudian dilakukan pengujian sesuai dengan path yang dihasilkan, yaitu 6 jalur pada aplikasi. Berdasarkan hasil uji tidak ditemukan galat dan berhasil.

4. KESIMPULAN

Berdasarkan pengujian yang dilakukan dengan *white box testing* pada aplikasi *daily activity monitoring* tidak ditemukan galat atau *error* dan dapat aplikasi berjalan dengan baik. *White box testing* dapat diterapkan pada aplikasi *daily activity monitoring*. Pengujian dilakukan pada fungsi simpan, ubah, hapus, dan tampil data. Semua fungsi dapat berjalan dengan baik disesuaikan alur program yang diujikan.

DAFTAR PUSTAKA

- [1] Syamsiah, S., Amelia, T., & Wulandari, S. H. E. (2015). Rancang Bangun Aplikasi Penilaian Kinerja Pegawai Berdasarkan Prestasi Kerja Berbasis Web. *Jsika*, 4(2), 1–6.
- [2] Aisyah, I., & Saifudin, A. (2020). Kenaikan Jabatan Menggunakan Fuzzy MADM Metode TOPSIS (Studi Kasus : PIZZA HUT Bintaro Plaza). *Jurnal Teknologi Informasi ESIT*, XV, 22–27.
- [3] Pressman, R. S. (2010). *Software Engineering : A Practitioner's Approach* (7th. Edition). New York: McGraw-Hill Higher Education.
- [4] Pare, S. (2013). Desain dan Implementasi E-commerce Pada Toko AS 88 Celluler Merauke. *Jurnal Ilmiah Mustek Anim Ha Vol.2*, 2(3), 222–229.

- [5] Subariah, R., & Permana, B. A. (2020). Perancangan Aplikasi Sales Support Untuk Manajemen Pelanggan Berbasis Web. *Jurnal Teknologi Informasi ESIT*, XV, 7–12. <http://jurnal-eresha.ac.id/index.php/esit/article/view/178/136>
- [6] Siswopranoto, A., & Mulyoto, A. (2020). Perancangan Game Edukasi Bahasa Inggris Untuk SD Menggunakan Eclipse. *Jurnal Teknologi Informasi ESIT*, XV, 1–6.
- [7] Ikhsan, A., Ma'mun, S., & Perdananto, A. (2020). Aplikasi Pemesanan Tiket Seminar Online Berbasis Web Di Universitas Pamulang. *Jurnal Teknologi Informasi ESIT*, XV, 13–21.
- [8] Saputri, G. (2020). Perancangan Sistem Informasi Rincian Biaya Produk Berbasis Web Pada Pt . Abc. *Jurnal Teknologi Informasi ESIT*, XV, 41–48.
- [9] Eriana, E. sita. (2020). Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Dengan White Box Testing. *Jurnal Teknologi Informasi ESIT*, XV, 28–33.