**VOL. IX NO. 2 AGUSTUS 2023**

JURNAL TEKNIK INFORMATIKA STMIK ANTAR BANGSA

Implementasi *Test Driven Development* dalam Pengembangan Sistem Informasi Berbasis Web untuk Data Penerima Bantuan Rumah Swadaya

Sigit Wijanarko1

***Abstract*— The existence of an information system that supports the activities of controlling, monitoring and evaluating the implementation of the DAK Sub-Bidang Rumah Swadaya can assist stakeholders in reviewing the profiles of beneficiaries and the progress of the self-help housing assistance stages. A software is said to be ready for use when it has gone through the testing phase. Test Driven Development (TDD) is a software development method that is carried out by making unit tests first before implementing program code. TDD and traditional development methods have different adaptations to change. Traditional development methods tend to have difficulty adapting to the changes that occur during development because each stage must be completed before starting the next. Meanwhile, TDD is more adaptable to changes because the development cycle continues to repeat, allowing improvements and changes at every stage. The application developed is in the form of an information system for the profile of beneficiaries and the progress of the stages of DAK self-help housing assistance in the PUPR Self-help Housing Sub- sector. The programming language used is PHP, while the tool used for testing is PHPUnit. The results of the TDD implementation prove that the functions of the built web application can work properly. In addition, the resulting program code is also organized and easy to read by other developers, making refactoring easier.**

***Intisari*—Adanya sistem informasi yang mendukung kegiatan pengendalian, pemantauan dan evaluasi pelaksanaan DAK Sub- Bidang Rumah Swadaya dapat membantu pihak pemangku kepentingan dalam mengulas profil penerima bantuan dan progres dari tahapan bantuan rumah swadaya. Sebuah perangkat lunak dikatakan siap untuk dipakai apabila sudah melalui tahap pengujian. Test Driven Development (TDD) adalah suatu metode pengembangan perangkat lunak yang dilakukan dengan membuat unit test terlebih dahulu sebelum mengimplementasikan kode program. TDD dan metode pengembangan tradisional memiliki adaptasi perubahan yang berbeda. Metode pengembangan tradisional cenderung sulit beradaptasi dengan perubahan yang terjadi selama pengembangan karena tiap tahap harus selesai sebelum memulai tahap berikutnya. Sementara itu, TDD lebih mudah beradaptasi denganan perubahan karena siklus pengembangan yang terus berulang memungkinkan perbaikan dan perubahan pada setiap tahap. Aplikasi yang dikembangkan adalah berupa sistem informasi profil penerima bantuan dan progres tahapan bantuan rumah swadaya DAK Sub-Bidang Rumah Swadaya PUPR. Bahasa pemograman yang digunakan adalah PHP, sedangkan alat yang digunakan untuk pengujian adalah PHPUnit. Hasil implementasi TDD membuktikan bahwa fungsi-fungsi dari aplikasi web yang dibangun dapat bekerja dengan baik. Selain itu, kode program yang dihasilkan juga menjadi tertata dan mudah dibaca oleh pengembang lain sehingga memudahkan *refactoring*.**

***Kata Kunci*— perangkat lunak; pengujian; TDD; aplikasi web.**

*Program Studi Teknik Informatika, STMIK Antar Bangsa, Jl. HOS Cokroaminoto, Kawasan Bisnis CBD Ciledug, Blok A5 No.29-36, Karang Tengah Ciledug, Kota Tangerang 15157 (telp:021-50686099; e-mail:* *sgtwijanarko23@gmail.com* *)*

1. PENDAHULUAN

Pelalsanaan DAK Sub-Bidang Rumah Swadaya memerlukan dukungan kegiatan pengendalian, pemantauan dan evaluasi sehingga dibutuhkan suatu system yang dapat meliputi hal tersebut. Pilihan metode pengembangan yang tepat berdasarkan pendekatan, fokus, waktu, adaptasi perubahan dan pemilik produk untuk system informasi tersebut adalah menggunakan TDD.

TDD adalah suatu pendekatan pengembangan perangkat lunak yang focus pada pengujian secara otomatis dalam setiap tahap penngembangan. Dalam TDD, pengembang menulis unit test terlebih dahulu sebelum menulis kode program actual. Unit test ini menguji fitur-fitur kecil atau modul-modul kode secara terisolasi untuk memastikan bahwa setiap bagian dari kode berfungsi dengan benar.

Berdasarkan metode TDD diharapkan dapat menghasilkan kode yang lebih baik, dengan memastikan bahwa setiap fitur atau modul telah diuji secara menyeluruh dan bekerja dengan baik sebelum menggabungkannya ke dalam sistem secara keseluruhan. Hail ini membantu mengurangi jumlah bug dan mempercepat proses pengembangan.

1. KAJIAN LITERATUR
2. *Test Driven Development*

TDD adalah suatu metode pengembangan perangkat lunak yang dilakukan dengan membuat unit test terlebih dahulu sebelum mengimplementasikan kode program.

Pada awalnya, pengembang membuat test case yang memeriksa sebuah fitur yang akan dibuat, dan baru kemudian mengimplementasikan kode program yang sesuai dengan test case tersebut. Setelah itu, pengembang menjalankan test case tersebut dan memastikan bahwas semua test case yang telah dibuat berhasil dijalankan.

Langkah-langkah umum yang dapat diikuti untuk mengimplementasikan TDD:

* 1. Tulis tes: Tulis tes yang merinci perilaku yang diinginkan dari kode. Pastikan gagal sebelum menulis kode, sehingga dapat yakin tes benar-benar menguji perilaku yang diinginkan.
	2. Jalankan tes: Jalankan tes untuk memastikan bahwa tes gagal karena kode belum ada.
	3. Tulis kode: Tulis kode yang memenuhi tes.
	4. Jalankan tes lagi: Jalankan tes lagi untuk memastikan bahwa kode memenuhi tes dan tidak ada tes yang gagal.
	5. Refaktor kode: Refaktor kode untuk memperbaiki keberlanjutan dan kualitas kode.
	6. Ulangi: Ulangi proses ini untuk setiap fitur atau fungsi yang dikembangkan.
1. *Perbedaan Antara TDD dan Metode Pengembangan Tradisional*

TDD dan metode pengembangan adalah dua pendekatan yang berbeda dalam pengembangan produk atau layanan dari segi:

* 1. Pendekatan: Metode pengembangan tradisional mengikuti pendekatan *waterfall* atau linear, yang berarti pengembangan dilakukan secara berurut, mulai dari perencanaan, desain, implementasi, pengujian, dan pemeliharaan. Sementara itu, TDD mengikuti pendekatan *agile*, yang berarti pengembangan dilakukan dalam iterasi singkat atau siklus pengembangan yang terus berulang.
	2. Fokus: Metode pengembangan tradisional lebih berfokus pada rencana dan dokumen, sedangkan TDD lebih berfokus pada interaksi dengan pelanggan dan umpan balik pengguna.
	3. Waktu: Metode pengembangan tradisional memerlukan waktu yang lebih lama untuk menyelesaikan pengembangan karena melibatkan banyak tahap yang harus dilakukan secara berurutan. Sedangkan TDD dapat lebih cepat karena pengembangan dilakukan dalam siklus singkat yang terus berulang.
	4. Perubahan: Metode pengembangan tradisional cenderung sulit beradaptasi dengan perubahan yang terjadi selama pengembangan karena setiap tahap harus selesai sebelum memulai tahap berikutnya. Sementara itu, TDD lebih mudah beradaptasi dengan perubahan karena siklus pengembangan yang terus berulang memungkinkan perbaikan dan perubahan pada setiap tahap.
	5. Pemilik produk: Metode pengembangan tradisional memiliki pemilik produk yang biasanya adalah manajer proyek, sementara pada TDD, pemilik produk adalah pihak bisnis atau pengguna yang memiliki perspektif lebih dalam terhadap produk atau layanan yang akan dikembangkan.
1. METODE PENELITIAN

Metodologi yang digunakan dalam penelitian ini adalah metodologi kualitatif. Dimana metodologi penelitian kualitatif adalah suatu metode yang digunakan untuk memahami fenomena tentang yang dialami oleh objek penelitian. Pada metodologi kualitatif tidak terdapat perhitungan matematika dan statistik. Metode TDD dalam peneliatan ini digunakan untuk menguji API data penerima bantuan.

Langkah pertama adalah membuat pengujian sesuai dengan proses bisnis dan desain dari sistem yang diinginkan dengan menyiapkan mocking untuk unit yang akan diuji (test case). Langkah kedua adalah menulis kode implementasi untuk unit dari mocking tersebut hingga lulus uji (code). Lalu langkah ketiga, kode implementasi unit yang sudah lulus uji dapat diperbaiki dan dirapihkan agar lebih mudah dipahami dan dikembangkan lebih lanjut (*refactor*), sebagaimana dapat dilihat pada Gambar 1.



Gambar 1. Alur Kerja TDD

Dengan menerapkan metode TDD, proses penggabungan unit akan menjadi lebih mudah karena setiap unit dipastikan bekerja dengan semestinya sebelum diintegrasikan satu sama lain. Manfaat dari menerapkan TDD adalah kemudahan untuk menemukan unit penyebab bug atau error dalam suatu sistem.

1. *Pengumpulan Data*

Metode pengumpulan data pada penelitian ini dilakukan dengan melakukan penarikan data langsung dari server simdak.perumahan.pu.go.id. Tujuan penarikan tersebut adalah untuk mendapatkan data tentang penerima bantuan DAK Ruswa.

1. *Implementasi*

Implementasi program dilakukan dengan menerapkan metode Test First Development (TFD). Adapun langkah- langkah pada metode TFD ditunjukan seperti gambar 2.



Gambar 2 Alur Kerja TFD

**VOL. IX NO. 2 AGUSTUS 2023**

JURNAL TEKNIK INFORMATIKA STMIK ANTAR BANGSA

Berdasarkan gambar 2, langkah pertama adalah membuat unit test hasil turunan dari user *requirement*. Saat pertama kali tes dijalankan hasilnya akan gagal/tidak berhasil, sebab belum terdapat kode apapun yang membuat tes menjadi terpenuhi. Oleh sebab itu, tugas pengembang adalah membuat kode sampai tes yang dibuat berhasil terpenuhi. Pengembang akan terus melakukan revisi kode program, dan apabila tes pertama sudah berhasil terpenuhi maka akan berganti ke tes-tes berikutnya sesuai fungsinya masing-masing. Apabila semua unit test sudah berhasil dijalankan, maka pengembangan sistem selesai dibangun.

1. *Refactoring*

Pada metode TFD, kode yang dihasilkan umumnya masih kurang rapi. Karena itu perlunya proses refactoring agar kode lebih mudah dipelihara dan berkelanjutan serta dipahami oleh pengembang lain. Proses *refactoring* dilakukan dengan memperhitungkan hasil pengujian agar tetap lulus tes artinya fitur atau modul tetap dapat bekerja sebagaimana mestinya (sesuai spesifikasi). Secara umum, TTD adalah gabungan antara TFD dengan refactoring.

1. HASIL DAN PEMBAHASAN

Keluaran dari penelitian ini menghasilkan sebuah aplikasi web yang digunakan untuk data penerima bantuan DAK Rumah Swadaya. Model pengembangan yang digunakan pada penelitian ini menggunakan metode *Test Driven Development*. Siklus pengembangan aplikasi yang dibangun dimulai dengan melakukan pengujian terlebih dahulu, kemudian melakukan implementasi program, dan terakhir melakukan refactoring.

1. *Pengujian Unit*

Proses pengujian unit dibuat dengan bantuan PHPUnit. Buat perpanjang kelas dasar PHPUnit sebagaimana gambar 3.

Gambar 3. buat perpanjangan kelas dasar PHPUnit Selanjutnya, tambahkan metode testPOST dan salin logika

POST dari skrip PenerimaBantuanTest.php:



Gambar 4. Penambahan metode testPOST

Terakhir, tambahkan beberapa penegasan untuk memeriksa bahwa kode statusnya adalah 201, bahwa kita memiliki tajuk Lokasi dan kita mendapatkan kembali JSON yang valid:



Gambar 5. Periksa status 201

1. *Implementasi Kode*

Pada tahap selanjutnya, API diterapkan kepada sistem berbasis web untuk tombol penambahan data penerima bantuan.



Gambar 6. Penerapan API pada tombol tambah penerima

1. *Refactoring*

Refactoring code adalah proses menstruktur ulang kode yang sudah diimplementasikan tanpa mengubah proses bisnisnya. Ini bertujuan untuk memproduksi kode yang mudah dibaca, mengurangi kompleksitas, memudahkan pemeliharaan dan tidak mengandung duplikasi.

Pada Gambar 7 menunjukkan bahwa Data Penerima Bantuan mengandung template yang dapat digunakan bersama oleh Pelaporan Progres Fisik dan Pelaporan Best Practice. Sehingga dapat meningkatkan readability dari potongan kode. Kemudian pada Tabel 1 menunjukkan bahwa hasil pengujian unit dengan menerapkan metode TDD pada tiap unit telah berhasil.



Gambar 7. Template yang dapat digunakan bersama

 Tabel 1. Hasil Pengujian Unit

|  |  |
| --- | --- |
|  Unit  | Hasil Pengujian  |
| Tambah Penerima Bantuan | Berhasil |
| Tambah Progres Bantuan | Berhasil |
|  Tambah Best Practice  | Berhasil  |

1. KESIMPULAN

Berdasarkan hasil implementasi dan perancangan dengan menerapkan metode TDD, dapat disimpulkan bahwa fitur-fitur dari sistem informasi berbasis web untuk data penerima bantuan rumah swadaya yang dibangun berjalan dengan baik. Dengan menerapkan metode TDD, pengembang dimudahkan dalam menemukan kesalahan dalam proses implementasi kode. Adanya proses *refactoring* dalam metode TDD menghasilkan kode yang lebih rapih setelah berhasil mengimplementasikan kode.

Terdapat juga kesulitan selama proses pengembangan. Diantara kesulitan dalam menerapkan metode TDD adalah memerlukan waktu dan biaya yang lebih karena harus menulis test case terlebih dahulu sebelum menulis kode program. Memerlukan skill dan pengalaman lebih karena harus mampu menulis test case yang baik dan benar pada proses mengubah kebutuhan dan proses bisnis yang diinginkan pengguna ke dalam bentuk test case.

UCAPAN TERIMA KASIH

Rasa syukur kepada Allah SWT, Tuhan semesta alam. Dan terima kasih diucapkan kepada semua pihak yang telah membantu, mendukung, serta membimbing selama penelitian ini dilakukan.

Sigit Wijanarko lahir di Sintang pada tanggal 23 Juni 1980. Lulus S1 Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Pembangunan Nasional – Jakarta pada tahun 2004. Lulus Magister Ilmu Komputer Program Pascasarjana Universitas Budi Luhur dengan konsentrasi Rekayasa Komputasi Terapan pada tahun 2019. Saat ini aktif sebagai Dosen tetap di STMIK Antar Bangsa dan praktisi IT di perusahaan swasta.

REFERENSI

1. Buffardi, K., & Edwards, S. H. *Impact of Teaching Test-Driven Development to Novice Programmers*. International Journal of Information and Computer Science IJICS, 1(6), 135-143. 2012.
2. Bulajic, A., Sambasivam, S., & Stojic, R. Overview of the Test Driven Development Research Projects and Experiments. Proceeding of the 2012 InSITE Conference, 2011, 165-187. [https://doi.org/10.28945/1647.](https://doi.org/10.28945/1647) 2012.
3. Ivo, A. A. S., Guerra, E. M., Porto, S. M., Choma, J., & Quiles, M. G. An Approach for Applying Test Driven Development (TDD) In the Development of randomized algorithms. Journal of Software Engineering Research and Development, 6(1). [https://doi.org/10.1186/s40411-018-00530-5.](https://doi.org/10.1186/s40411-018-00530-5) 2018.
4. Khanam, Z., & Ahsan, M. N. Evaluating the Effectiveness of Test Driven Development: Advantages and pitfalls. International Journal of Applied Engineering Research, 12(18), 7705-7716. 2017.
5. Moe, M. M. Comparative Study of Test Driven Development TDD, Behaviour-Driven Development BDD and Acceptance Test-Driven Development ATDD. International Journal of Trend in Science Research and Development, Volumen-3(issue-4), 231-234. [https://doi.org/10.31142/ijtsrd23698.](https://doi.org/10.31142/ijtsrd23698) 2019.
6. Muhammad Shahid Khan, N. K. M. A. K. M. A. J. Reducing Testing Effort in the Test Driven Development. Global Journal of Computer Science and Technology, 13(7), 0-4. 2013.
7. Rós Aguilar, R. Using Test-Driven Development to Improve Software Development Practices. 2016.