

Analisis dan Pengujian Perangkat Lunak Sistem Informasi Pembayaran Sekolah dengan Metode Pengukuran Kualitas SQuaRE

Rahman Abdillah^{1*}, Rudi Hermawan², Wawan Hermawansyah³, Dwi Puspita Agustin⁴,
Ibnu Adkha⁵, Nur Alam⁶

^{1,2}Program Studi Teknik Informatika, Universitas Indraprasta PGRI, Indonesia

³Program Studi Sistem Informasi, Universitas Bani Saleh, Indonesia

^{4,5,6}Program Studi Teknologi Rekayasa Multimedia, Politeknik Bhakti Kartini, Indonesia

Alamat Kampus: Jl. Nangka Raya No. 58 C, Tanjung Barat, Jagakarsa, Kota Jakarta Selatan

Korespondensi penulis: rabdil.bu@gmail.com*

Abstract. *Software quality testing is a crucial stage in the development of information systems to ensure their reliability and functionality. This study aims to analyze and test the software quality of the School Payment Information System using the Software Product Quality Requirements and Evaluation (SQuaRE) method. This method refers to the ISO/IEC 25000 standard, which measures various aspects of software quality, including functionality, reliability, efficiency, compatibility, usability, security, maintainability, and portability. The aim of this research was conducted through a series of tests by collecting data from system specifications, test designs, and user feedback. The test results indicate that the evaluated information system meets most of the quality indicators based on the SQuaRE standard, with some aspects still requiring improvement, particularly in terms of usability and system efficiency. This evaluation provides deeper insights into the effectiveness of applying the SQuaRE method in software testing and offers recommendations for improving system quality. Thus, the implementation of the SQuaRE method is expected to enhance the reliability and efficiency of the school payment information system, thereby supporting more effective and accurate financial management in schools.*

Keywords: *Software testing, SQuaRE, ISO/IEC 25000, school payment system, black-box testing.*

Abstrak. Pengujian kualitas perangkat lunak merupakan tahap krusial dalam pengembangan sistem informasi untuk memastikan keandalan dan fungsionalitasnya. Penelitian ini bertujuan untuk menganalisis dan menguji kualitas perangkat lunak pada Sistem Informasi Pembayaran Sekolah dengan menggunakan metode Software Product Quality Requirements and Evaluation (SQuaRE). Metode ini mengacu pada standar ISO/IEC 25000 yang mengukur berbagai aspek kualitas perangkat lunak, termasuk fungsionalitas, keandalan, efisiensi, kompatibilitas, kegunaan, keamanan, pemeliharaan, dan portabilitas. Tujuan dari penelitian ini dilakukan melalui serangkaian pengujian dengan mengumpulkan data dari spesifikasi sistem, desain pengujian, serta umpan balik pengguna. Hasil pengujian menunjukkan bahwa sistem informasi yang diuji telah memenuhi sebagian besar indikator kualitas sesuai standar SQuaRE, dengan beberapa aspek yang masih perlu diperbaiki, terutama dalam kemudahan penggunaan dan efisiensi sistem. Evaluasi ini memberikan wawasan yang lebih mendalam mengenai efektivitas penerapan metode SQuaRE dalam pengujian perangkat lunak serta rekomendasi perbaikan untuk meningkatkan kualitas sistem. Dengan demikian, implementasi metode SQuaRE diharapkan dapat membantu meningkatkan keandalan dan efisiensi sistem informasi pembayaran sekolah, sehingga mampu mendukung pengelolaan keuangan sekolah secara lebih efektif dan akurat.

Kata kunci: Pengujian perangkat lunak, SQuaRE, ISO/IEC 25000, sistem informasi pembayaran, black-box testing.

1. LATAR BELAKANG

Pada dewasa ini, pengembangan perangkat lunak seperti sistem informasi dan aplikasi merupakan aspek yang cukup penting dalam mendukung berbagai aktivitas di era digital. Sistem informasi yang dirancang dengan baik memungkinkan organisasi untuk dapat mengelola data, meningkatkan efisiensi operasional, serta mendukung pengambilan keputusan yang lebih akurat. Seiring dengan perkembangan teknologi, berbagai macam metode dan

pendekatan dalam pengembangan perangkat lunak terus berkembang, mulai dari model tradisional seperti waterfall, SDLC hingga pendekatan yang lebih adaptif seperti agile dan DevOps. Keberhasilan suatu sistem informasi tidak hanya bergantung pada fungsionalitasnya, tetapi juga pada aspek kualitas seperti keandalan, keamanan, dan kemudahan penggunaan.

Dalam proses pengembangannya, perangkat lunak harus memenuhi kebutuhan pengguna serta mempertimbangkan faktor-faktor seperti skalabilitas, interoperabilitas, dan efisiensi sumber daya. Standar kualitas seperti ISO/IEC 25000 (*Software Product Quality Requirements and Evaluation* – SQuaRE) sering digunakan sebagai acuan dalam menilai kualitas perangkat lunak yang dikembangkan. Selain itu, penerapan teori terbaru dalam pengembangan rekayasa perangkat lunak, seperti manajemen proyek yang efektif dan pengujian berkelanjutan, sangat diperlukan untuk memastikan bahwa sistem yang dihasilkan dapat beroperasi secara optimal dalam berbagai skenario penggunaan.

Penelitian ini bertujuan untuk menguji kualitas perangkat lunak dalam studi kasus berupa sistem informasi pembayaran sekolah menggunakan metode SQuaRE dengan fokus pada beberapa karakteristik utama yang ditentukan dalam standar ISO. Pengujian dilakukan dengan cara mengukur parameter-parameter yang relevan serta menganalisis sejauh mana perangkat lunak yang diuji memenuhi kriteria kualitas yang ditetapkan. Dengan pendekatan ini, diharapkan dapat dihasilkan suatu evaluasi dan rekomendasi yang komprehensif guna memberikan wawasan bagi para pengembang perangkat lunak dalam meningkatkan kualitas produk mereka.

Artikel ini disusun sebagai kontribusi terhadap pengembangan metode evaluasi kualitas perangkat lunak yang lebih sistematis dan berbasis standar. Struktur artikel ini diawali dengan pembahasan kajian teoritis terkait tahap-tahap pengembangan perangkat lunak (SDLC), pengujian kualitas perangkat lunak dan metode SQuaRE, dilanjutkan dengan metodologi penelitian yang digunakan. Selanjutnya, hasil pengujian dan analisis disajikan untuk memberikan gambaran mengenai efektivitas metode SQuaRE dalam menilai kualitas perangkat lunak. Bagian terakhir dari artikel ini berupa kesimpulan serta rekomendasi untuk penelitian lebih lanjut.

2. KAJIAN TEORITIS

Perangkat lunak yang berkualitas menjadi salah satu faktor utama dalam menentukan keberhasilan dari pengembangan suatu sistem informasi atau aplikasi. Seiring dengan meningkatnya ketergantungan manusia terhadap teknologi dalam berbagai bidang, pengujian kualitas perangkat lunak menjadi aspek yang sangat krusial. Standar internasional seperti

Software Product Quality Requirements and Evaluation (SQuaRE) dikembangkan untuk memberikan pendekatan sistematis dalam menilai kualitas dari perangkat lunak. Pendekatan ini membantu memastikan bahwa perangkat lunak memenuhi kebutuhan pengguna, memiliki performa optimal, serta dapat diandalkan dalam beberapa kondisi operasional di lapangan. Dibawah ini merupakan kajian teoritis dan landasan teori terkait penelitian pada artikel ini:

SDLC (System Development Life Cycle)

Sistem Development Life Cycle (SDLC) atau disebut juga dengan Siklus Hidup Pengembangan Sistem merupakan suatu metode tahapan yang sering digunakan dalam pengembangan perangkat lunak. Secara historis, konsep SDLC mulai dikembangkan sejak era 1960-an seiring dengan meningkatnya kebutuhan akan perangkat lunak yang lebih kompleks dan dapat diandalkan. Salah satu tokoh yang dikenal dalam pengembangan model SDLC adalah Winston W. Royce, yang pada tahun 1970 memperkenalkan konsep Waterfall Model sebagai pendekatan awal dalam pengembangan perangkat lunak (Sommerville, 2016). Seiring berjalannya waktu, model ini dikembangkan lebih lanjut oleh berbagai ahli dalam bidang rekayasa perangkat lunak dan sistem informasi. Landasan teori SDLC berakar pada prinsip-prinsip rekayasa perangkat lunak, seperti modularitas, pemisahan tugas, dan siklus iteratif untuk memastikan bahwa sistem dapat dikembangkan secara efisien dan sesuai dengan kebutuhan pengguna.

System Development Life Cycle (SDLC) adalah sebuah kerangka kerja sistematis yang digunakan dalam pengembangan perangkat lunak dan sistem informasi. SDLC mencakup serangkaian tahapan mulai dari analisis kebutuhan, perancangan, pengembangan, pengujian, implementasi, hingga pemeliharaan sistem (Sommerville, 2016). Model ini membantu memastikan bahwa perangkat lunak dikembangkan secara terstruktur dan memenuhi standar kualitas serta kebutuhan pengguna. Beberapa pendekatan yang umum digunakan dalam SDLC antara lain *Waterfall Model*, *V-Model*, *Iterative Model*, *Spiral Model*, serta metode yang lebih fleksibel dan modern seperti Agile dan DevOps (Pressman, 2015). Dengan mengikuti tahapan dalam SDLC, organisasi dapat meminimalisir risiko kesalahan serta meningkatkan efisiensi dalam pengembangan sebuah sistem perangkat lunak.

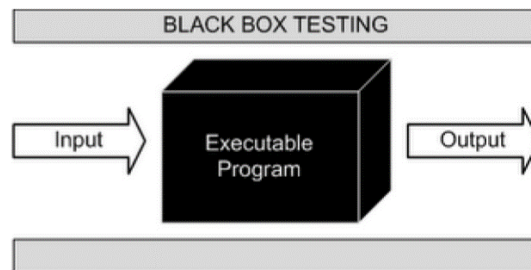
Pengujian Perangkat Lunak

Pengujian perangkat lunak (*software testing*) merupakan salah satu tahapan krusial dalam *System Development Life Cycle* (SDLC) yang bertujuan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Tahapan ini

bertujuan untuk mengidentifikasi kesalahan (*bugs*), memastikan keandalan sistem, serta mengevaluasi kinerja dan keamanan perangkat lunak sebelum diterapkan ke lingkungan produksi. Secara umum, pengujian perangkat lunak dibagi menjadi dua kategori utama, yaitu *verification* dan *validation* (Jorgensen, 2014). *Verification* memastikan bahwa perangkat lunak dikembangkan dengan benar sesuai dengan desain dan spesifikasi, sementara *validation* memastikan bahwa perangkat lunak yang dihasilkan memenuhi kebutuhan pengguna. Metode pengujian dapat dilakukan secara manual maupun otomatis dengan menggunakan berbagai alat bantu pengujian (*testing tools*).

Kajian teori mengenai pengujian perangkat lunak berakar pada berbagai standar dan metodologi yang telah dikembangkan dalam rekayasa perangkat lunak. IEEE 829 dan ISO/IEC 25000 (SQuaRE) adalah beberapa standar yang banyak digunakan dalam mendokumentasikan dan mengevaluasi proses pengujian perangkat lunak. Teknik pengujian dapat dibagi menjadi beberapa jenis, seperti *black-box testing*, yang berfokus pada pengujian fungsional tanpa mengetahui kode sumber, *white-box testing*, yang menguji logika internal dan struktur kode, serta kombinasi dari *black-box* dan *white-box testing* disebut dengan *grey-box testing* (Pebrianto, 2020). Selain itu, terdapat berbagai tingkat pengujian dalam SDLC, termasuk *unit testing*, *integration testing*, *system testing*, dan *acceptance testing*, yang masing-masing memiliki peran dalam memastikan bahwa perangkat lunak bekerja dengan baik sebelum dirilis kepada pengguna akhir.

Metode *black-box* berfokus pada pengujian aspek fungsional dari perangkat lunak, sedangkan metode *white-box* mengevaluasi struktur internal program. Masing-masing metode memiliki kelebihan dan kekurangannya sendiri (Muslimin, 2020). Sebagai contoh, metode *black-box*, yang mengutamakan pengujian berdasarkan fungsionalitas dan masukan (*input*), dapat menghadapi kendala dalam mendokumentasikan hasil pengujian jika spesifikasi perangkat lunak kurang jelas. Di sisi lain, metode *white-box*, yang lebih menitikberatkan pada analisis struktur kode program, memerlukan sumber daya yang lebih besar, terutama saat diterapkan pada perangkat lunak berskala besar. Selain itu, metode ini juga mengharuskan penguji memiliki pemahaman mendalam terhadap kode program yang diuji. Meskipun demikian, pengujian berbasis struktur dianggap memiliki peran penting karena lebih efektif dalam menekan biaya pengujian perangkat lunak dibandingkan dengan pengujian berbasis fungsional (Sulistyanto, 2014).

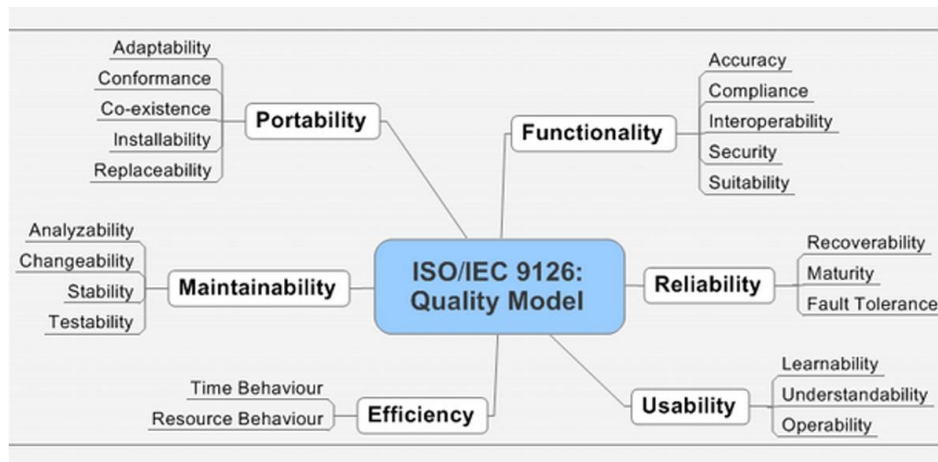


Gambar 1. Pengujian Perangkat Lunak dengan Metode Black Box Testing

Metode *black-box testing* lebih banyak digunakan dibandingkan metode pengujian lainnya karena fokusnya pada pengujian fungsional tanpa memerlukan pemahaman terhadap kode sumber perangkat lunak. Teknik ini memungkinkan penguji untuk mengevaluasi sistem berdasarkan masukan (*input*) dan keluaran (*output*), sehingga lebih fleksibel dan dapat diterapkan oleh berbagai pihak, termasuk tim pengujian independen maupun pengguna akhir. Selain itu, *black-box testing* dapat digunakan pada berbagai tahap pengembangan perangkat lunak, mulai dari *unit testing* hingga *system testing* dan *acceptance testing*. Keunggulan lainnya adalah kemampuannya dalam mengidentifikasi kesalahan pada antarmuka pengguna dan interaksi sistem dengan lingkungan eksternal tanpa harus menganalisis detail implementasi kode. Oleh karena itu, metode ini lebih sering diterapkan dalam pengujian perangkat lunak, terutama pada proyek yang memiliki batasan waktu dan sumber daya yang terbatas.

Standar Kualitas SQuaRE

Metode *Software Product Quality Requirements and Evaluation* (SQuaRE) merupakan standar internasional yang dikembangkan oleh ISO/IEC 25000 untuk mengevaluasi kualitas perangkat lunak secara sistematis. Metode ini mengelompokkan kualitas perangkat lunak ke dalam berbagai karakteristik utama, seperti fungsionalitas, keandalan, efisiensi kinerja, kompatibilitas, kegunaan, keamanan, dan pemeliharaan. Pendekatan SQuaRE memastikan bahwa perangkat lunak tidak hanya berfungsi sesuai spesifikasi, tetapi juga memenuhi standar kualitas yang dapat meningkatkan pengalaman pengguna dan keberlanjutan sistem. Selain itu, metode ini mencakup proses pengukuran kualitas berdasarkan metrik yang telah ditentukan, sehingga memungkinkan pemangku kepentingan untuk mengevaluasi dan meningkatkan perangkat lunak secara objektif.



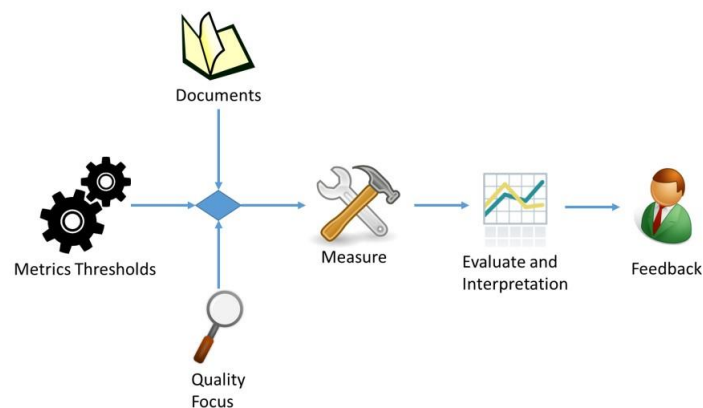
Gambar 2. ISO/IEC 9126: Quality Model

Pengujian kualitas dengan metode SQuaRE dari gambar diatas dapat dilihat dari berbagai aspek atau indikator, yakni: *Maintainability* (pemeliharaan), *Efficiency* (efisiensi), *Usability* (kegunaan), *Reliability* (keandalan), *Functionality* (fungsionalitas), dan *Portability* (kemudahan dalam pemindahan). Pada berbagai aspek kualitas tersebut, pengujian perangkat lunak dilakukan melalui beberapa tahap, mulai dari identifikasi kebutuhan, spesifikasi metrik pengukuran, hingga analisis hasil evaluasi. Standar ini memberikan kerangka kerja yang dapat disesuaikan dengan kebutuhan proyek, baik dalam pengembangan perangkat lunak baru maupun peningkatan sistem yang sudah ada. Salah satu keunggulan utama SQuaRE adalah kemampuannya dalam mengintegrasikan proses evaluasi ke dalam siklus pengembangan perangkat lunak (*Software Development Life Cycle*), sehingga pengujian kualitas dapat dilakukan secara berkelanjutan. Dengan pendekatan berbasis standar ini, organisasi dapat memastikan bahwa perangkat lunak yang dikembangkan tidak hanya memenuhi kebutuhan teknis, tetapi juga memberikan nilai tambah bagi pengguna dan bisnis secara keseluruhan.

3. METODE PENELITIAN

Metode *Black-Box Testing* merupakan teknik pengujian perangkat lunak yang bertujuan untuk mendeteksi kesalahan dalam perancangan sistem, seperti malfungsi pada fitur, kesalahan pada menu, atau ketidaksesuaian dalam fungsionalitas (Uminingsih, 2022). Pengujian ini lebih cocok digunakan untuk mengevaluasi fungsionalitas perangkat lunak tanpa memeriksa struktur internal atau kode sumbernya. Dalam proses pengujian, berbagai data masukan diuji secara acak untuk melihat respons sistem terhadap input yang diberikan. Jika input yang dimasukkan tidak valid, sistem akan menolaknya atau tidak menyimpannya dalam basis data, sedangkan input yang valid akan diterima dan diproses sesuai dengan aturan sistem informasi.

Terdapat beberapa teknik dalam *Black-Box Testing*, di antaranya: (a) *Equivalence Partitioning*, yaitu membagi data masukan ke dalam beberapa partisi agar pengujian lebih efisien; (b) *Boundary Value Analysis*, yang berfokus pada pengujian nilai batas minimum dan maksimum untuk mendeteksi kesalahan di luar atau dalam perangkat lunak; (c) *Fuzzing*, yang bertujuan menemukan bug dengan menyuntikkan data tidak valid ke dalam sistem; (d) *Cause-Effect Graph*, yang menggunakan diagram untuk menggambarkan hubungan sebab-akibat dalam pengujian; (e) *Orthogonal Array Testing*, yang digunakan untuk menguji domain input kecil namun kompleks pada skala besar; (f) *All Pair Testing*, yaitu metode yang menguji semua kombinasi pasangan parameter input untuk memastikan setiap kombinasi diuji; serta (g) *State Transition*, yang berfungsi untuk menguji kondisi dan navigasi sistem dalam bentuk grafik (Beizer, 1990). Dalam penelitian ini, teknik yang digunakan untuk menguji Sistem Informasi Pembayaran Sekolah adalah *Equivalence Partitioning*. Teknik ini membagi input dari unit perangkat lunak ke dalam beberapa partisi, dari mana kasus uji dapat diturunkan. Pendekatan ini bertujuan untuk mengidentifikasi kelas kesalahan sehingga jumlah kasus uji dapat diminimalkan tanpa mengurangi cakupan pengujian (Jaya et al., 2019).

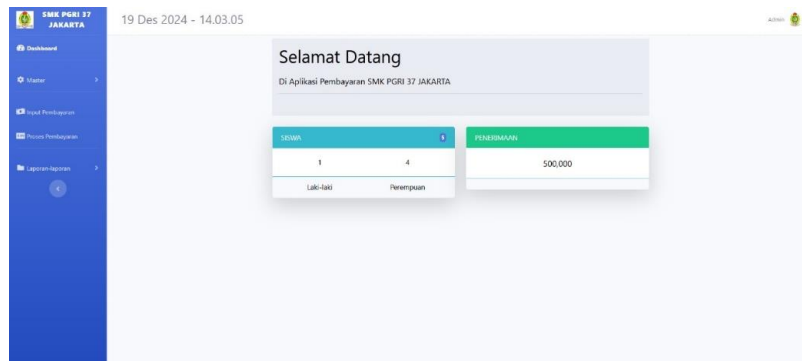


Gambar 3. Prosedur untuk pengukuran kualitas produk

4. HASIL DAN PEMBAHASAN

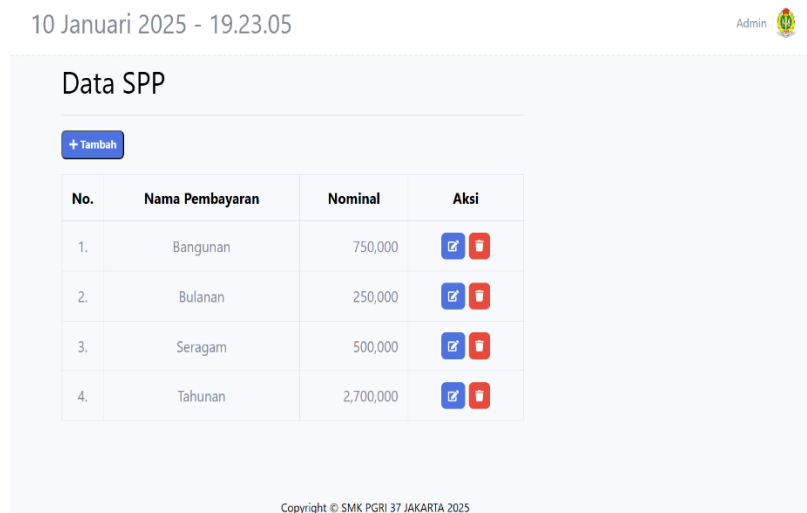
Pada artikel ini penulis menggunakan teknik pengujian dengan metode *black box testing* dengan teknik pengujian *Equivalence Partitioning*. Dibawah ini merupakan contoh sistem informasi pembayaran sekolah. Sistem informasi ini masih bersifat prototipe dan belum digunakan pada sekolah tersebut. Dari gambar *dashboard* dibawah ini, dapat kita lihat beberapa fitur atau menu layanan sistem informasi, seperti data master, data transaksi, dan menu laporan.

ANALISIS DAN PENGUJIAN PERANGKAT LUNAK SISTEM INFORMASI PEMBAYARAN SEKOLAH DENGAN METODE PENGUKURAN KUALITAS SQUARE



Gambar 4. Tampilan Dashboard Sistem Informasi Pembayaran Sekolah

Penulis melakukan pengujian secara acak terhadap berbagai fitur dalam sistem informasi pembayaran sekolah. Pengujian tahap pertama yang dilakukan penulis adalah aspek *usability* (kegunaan). Salah satu pengujian dilakukan pada menu data pembayaran dengan memasukkan beberapa data, seperti no, nama pembayaran, nominal, dan aksi. Namun, setelah dilakukan pemeriksaan lebih lanjut, ditemukan bahwa tidak terdapat kolom khusus yang menunjukkan nama dan tanggal pembayaran, sehingga dari data pelaporan pembayaran belum lengkap. Untuk koneksi kedalam database SQL sudah dapat beroperasi dengan benar, sehingga secara fungsionalitas sudah bisa digunakan.



Gambar 5. Tampilan Data Pembayaran Siswa

Sistem informasi dapat melakukan proses cetak pelaporan data dengan menghasilkan file dengan format *.pdf. Data yang tampil pada file pelaporan juga sudah sesuai dengan data yang terdapat pada tampilan data pembayaran spp. Hasil dari cetak pelaporan dapat dilihat pada gambar 6.

Pada tahap pengujian selanjutnya, penulis melakukan pengujian terhadap aspek *portability* (kemudahan dalam pemindahan). Pada tahap ini, penulis mencoba untuk mengirimkan file melalui email dan whatsapp untuk pemindahan perangkat lunak (*software*) dan di install pada perangkat komputer yang baru. Hasilnya, aplikasi sistem informasi tersebut dapat digunakan pada perangkat yang baru, dengan tahapan yang sama seperti menggunakan xampp sebagai localhost server.

5. KESIMPULAN DAN SARAN

Pengujian perangkat lunak bertujuan untuk memastikan kualitas produk perangkat lunak yang dihasilkan. Dengan menerapkan metode pengujian yang menyeluruh, perusahaan atau pengguna dapat memastikan bahwa sistem informasi inventori barang dapat beroperasi secara optimal dan sesuai dengan kebutuhan. Salah satu pendekatan yang dapat digunakan adalah metode Software Product Quality Requirements and Evaluation (SQuaRE), yang merupakan standar ISO/IEC 25000 untuk mengevaluasi kualitas perangkat lunak berdasarkan berbagai aspek, seperti fungsionalitas, keandalan, efisiensi, dan keamanan. Secara umum, terdapat dua jenis utama dalam pengujian perangkat lunak, yaitu *white-box testing* dan *black-box testing*. Setiap metode pengujian memiliki keunggulan dan keterbatasannya masing-masing, sehingga pemilihan metode yang tepat harus mempertimbangkan faktor-faktor seperti efisiensi sumber daya, batas waktu, kebutuhan pengguna atau klien, serta kesiapan infrastruktur dan teknologi yang tersedia. Dengan mengadopsi SQuaRE, evaluasi kualitas perangkat lunak dapat dilakukan secara lebih sistematis dan terukur.

Tahap pengujian dalam pengembangan perangkat lunak memiliki peran yang sangat penting untuk mencegah kesalahan dalam pengelolaan sistem informasi pembayaran sekolah, yang berpotensi menyebabkan kerugian finansial atau gangguan dalam pendataan administrasi. Pengujian yang dilakukan secara optimal juga dapat meningkatkan kepercayaan pengguna terhadap sistem, karena mereka dapat mengandalkan akurasi data yang dihasilkan untuk mendukung pengambilan keputusan bisnis yang lebih baik. Oleh karena itu, investasi dalam proses pengujian perangkat lunak dengan metode SQuaRE merupakan langkah strategis yang mendukung keberhasilan implementasi sistem informasi pembayaran sekolah secara keseluruhan.

DAFTAR REFERENSI

- Beizer, B. (1990). *Software Testing Techniques 2nd Edition*. Boston: International Thompson Computer Press.
- Jaya, M. S., Gumilang, P., Wati, T., Andersen, Y. P., & Desyani, T. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 131-136. <https://doi.org/10.32493/informatika.v4i4.3834>
- Jorgensen, P. C. (2014). *Software Testing 4th Edition*. Florida: Taylor & Francis Group.
- Muslimin, D. B., Kusmanto, D., Amelia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19-26. <https://doi.org/10.32493/informatika.v5i1.3778>
- Pebrianto, A., Abdillah, R., Valentino, V. H. (2020). Perancangan Sistem Informasi Penjualan Donat pada PT.King Alianz Donuts. *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, 01(03), 424-429. <https://doi.org/10.30998/jrami.v1i03.324>.
- Pressman, R. (2015). *Software Engineering: A Practitioner Approach 8th Edition*. New York : McGraw Hill.
- Setiawan, R. (2017). Pengujian Perangkat Lunak. *Seminar Nasional Informatika dan Aplikasinya (SNIA)* 2017. (c37-c39). http://repository.unjani.ac.id/index.php?p=show_detail&id=561&keywords=
- Sommerville, Ian. (2016). *Software Engineering 10th Edition*. London: Pearson Education Limited.
- Sulistiyanto, H & Azhari, S.N. (2014). Urgensi Pengujian Pada Kemajemukan Perangkat Lunak dalam Multi Perspektif. *KomuniTi*. 01(01), 65-74 <https://doi.org/10.23917/komuniti.v6i1.2944>
- Uminingsih, Ichsanudin, M. N., Yusuf, M. Suraya. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan dengan Metode Black Box Testing bagi Pemula. *Jurnal Ilmiah Teknik dan Ilmu Komputer (STORAGE)*, 01(02). 1-8
- Wahyu, M. T., Afrizal, M,. (2023). Pengujian Blackbox Metode Equivalent Partitions pada Aplikasi Karyawan Website Oby Komputer. *Jurnal Sistem Informasi (Teknofile)*, 01(02). <https://jurnal.nawansa.com/index.php/teknofile/article/view/16/12>
- Wardhani, D. R., & Abdillah, R. (2018). Pengambilan Keputusan dengan Metode Perbandingan Eksponensial (MPE) dalam Manajemen Kedai. *Semnasristek 2018*, (pp. 439-444). <https://osf.io/preprints/osf/rx2p5>