

PENGEMBANGAN APLIKASI REKAPITULASI KEUANGAN BERBASIS DESKTOP PROGRAM STUDI TEKNOLOGI INFORMASI UMY

(PENGEMBANGAN APLIKASI REKAPITULASI KEUANGAN BERBASIS DEKSTOP PROGRAM STUDI TEKNOLOGI INFORMASI UMY)

TRI YOGA UTAMA

ABSTRACT

Information technology in its use is used to manage an information process in an institution, organization or institution. The need for effectiveness and time efficiency causes very important application of information technology. No exception is the financial recap process at UMY's Information Technology Study Program. The problem faced is that the financial recapitulation process that is still carried out can still cause a risk of inaccurate data. The Information Technology Study Program requires an application that can reduce these risks. Then a desktop based application is developed using the Swing Framework, with SQL Server 2017 database, the IDE used is Netbeans 8.1, iReport for the printing process, and using the Java programming language. The research method used is SDLC (System Development Life Cycle). SDLC (Systems Development Life Cycle) is a systematic approach that divides the software development process into various phases, such as requirements, design, and coding.

Based on the test results it can be concluded that the application built can help the process of recapitulating financial data more quickly, and can reduce the risk of data inaccuracies.

Keywords: financial recapitulation, desktop, swing framework

I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini sudah sedemikian pesat dan menjangkau hampir seluruh aspek dalam bisnis manapun di dunia. Di antara berbagai bisnis yang ada saat ini. Dengan teknologi informasi, kita dapat dengan mudah mendapatkan informasi yang dibutuhkan, serta memiliki peranan yang besar dalam pelayanan berbasis teknologi informasi.

Teknologi informasi dalam pemanfaatannya digunakan untuk mengelola suatu proses informasi dalam suatu lembaga, organisasi maupun institusi. Kebutuhan efektifitas dan efisiensi waktu menyebabkan sangat penting menerapkan teknologi informasi. Tak terkecuali proses rekap keuangan di Program Studi Teknologi Informasi UMY. Permasalahan yang dihadapi adalah proses perekapan keuangan yang dilakukan masih dapat menyebabkan resiko

ketidakakuratan data serta menyebabkan kesulitan dalam hal pengawasan data keuangan.

Untuk mengatasi masalah tersebut, maka perlu dikembangkan sebuah aplikasi rekap keuangan berbasis desktop. Dengan adanya aplikasi ini diharapkan dapat meminimalisir resiko – resiko yang ada.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas dapat diidentifikasi masalah sebagai berikut:

1. Kelemahan dalam proses pengolahan informasi yang ada, sehingga terjadi ketidakakuratan informasi yang berpengaruh terhadap laporan untuk pihak fakultas.
2. Kesulitan dalam melakukan *monitoring* aliran dana yang ada dalam prodi.

1.3 Rumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan permasalahan yang akan diselesaikan dalam penelitian ini:

1. Bagaimana agar dapat mengurangi resiko ketidakakuratan dalam proses perekapan data keuangan ?
2. Bagaimana agar proses rekap keuangan dapat dikelola dengan lebih baik serta dapat melakukan *monitoring* menggunakan aplikasi desktop?

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk merancang aplikasi untuk proses pengolahan data keuangan di Program Studi Teknologi Informasi UMY.

1.5 Batasan Masalah

Berdasarkan latar belakang diatas didapatkan batasan masalah sebagai berikut:

1. Aplikasi dekstop dirancang hanya untuk *admin*.
2. *Database* yang digunakan masih menggunakan *database* lokal.
3. Fitur cetak hanya berdasarkan tahun ajaran.

1.6 Manfaat Penelitian

Manfaat penelitian ini adalah:

1. Meningkatkan efisiensi waktu dan kinerja melalui basis data yang lebih terstruktur.
2. Pemanfaatan teknologi aplikasi dekstop agar proses rekap keuangan dapat maksimal dan lebih akurat.

II. LANDASAN TEORI

2.1 Laporan Keuangan

Menurut (Kasmir, 2008) Laporan keuangan adalah laporan yang menunjukkan kondisi keuangan perusahaan pada saat ini atau pada periode tertentu. Menurut Standar Akuntansi Keuangan yang dikeluarkan oleh Ikatan Akuntan Indonesia tujuan laporan keuangan adalah menyediakan informasi yang menyangkut posisi keuangan, kinerja, serta perubahan posisi keuangan suatu perusahaan yang bermanfaat Karakteristik kualitatif merupakan ciri khas yang membuat informasi dalam laporan keuangan berguna bagi pemakai.

Terdapat empat karakteristik kualitatif pokok yaitu :

- Dapat Dipahami

Informasi yang disajikan dalam laporan keuangan dapat dipahami peserta dan bentuk serta istilahnya disesuaikan dengan batas para pengguna.

- Relevan

Laporan keuangan dianggap jika informasi yang disajikan didalamnya dapat mempengaruhi keputusan pengguna.

- Keandalan

Informasi dalam laporan keuangan bebas dari pengertian yang menyesatkan dan kesalahan material;

- Dapat diperbandingkan

Informasi yang disajikan akan lebih berguna bila dapat diperbandingkan dengan laporan keuangan pada periode sebelumnya. bagi sejumlah besar pemakai dalam pengambilan keputusan.

2.2 Aplikasi Desktop

Berdasarkan buku yang ditulis oleh (Mudakir, 2013) aplikasi desktop adalah aplikasi yang dapat berjalan sendiri tanpa menggunakan browser atau internet di dalam komputer tersebut.

2.3 Unified Modeling Language (UML)

Menurut (Sri & Wahono, 2003) *Unified Modeling Language* (UML) merupakan bahasa yang telah menjadi standar dalam industri untuk merancang dan mendokumentasi sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Model UML yang dipakai dalam pengembangan aplikasi yaitu model *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*.

a. Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login*, *mem-cretate* sebuah bukti transaksi, dan sebagainya. Sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

b. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

c. *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

2.4 Entity Relationship Diagram (ERD)

Menurut (A. Permana, 2017) *Entity Relationship Diagram* (ERD) adalah suatu model yang menggambarkan hubungan antar data dalam *database* berdasarkan objek-objek dasar data yang mempunyai relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkan digunakan beberapa notasi dan simbol. Ada tiga komponen dasar yang digunakan dalam permodelan ERD, yaitu :

a. Entitas

Entitas atau entiti adalah objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut.

c. Relasi

Relasi adalah hubungan antara beberapa entitas yang berasal dari himpunan entitas yang berbeda. Ada beberapa relasi dalam ERD, *one to one*, *one to many*, dan *many to many*.

2.5 System Development Life Cycle (SDLC)

Menurut (B. Permana, 2017) SDLC (*Systems Development Life Cycle*) adalah sebuah pendekatan sistematis yang membagi proses pengembangan perangkat lunak ke dalam berbagai fase, seperti persyaratan, desain, dan pengkodean. SDLC membantu

untuk melihat perkiraan biaya, jadwal, dan kualitas proyek perangkat lunak. SDLC membagi pengembangan perangkat lunak dalam 6 fase, yaitu:

- Analisis Kelayakan (*feasibility analysis*).
- Analisis kebutuhan dan spesifikasi (*Requirement analysis and specification*).
- Desain (*design*).
- Pengodean (*coding*).
- Pengujian (*testing*).
- Pemeliharaan (*maintenance*).

Dalam membangun sebuah aplikasi, setiap jenisnya memiliki persyaratan yang berbeda-beda. Hal ini diperlukan untuk menyesuaikan fase SDLC memenuhi kebutuhan yang lebih spesifik dari *web* atau aplikasi. Dalam proses penyesuaian fase SDLC memunculkan berbagai pendekatan pengembangan perangkat lunak. Berikut pendekatan-pendekatan dalam SDLC:

- Waterfall approach*, yaitu pendekatan yang menjelaskan proses pengembangan perangkat lunak dalam aliran linier berurutan.
- Prototyping approach*, yaitu pendekatan yang juga dikenal sebagai pendekatan evolusioner.
- Spiral approach*, yaitu pendekatan untuk mengembangkan perangkat lunak yang dirilis dalam berbagai versi.
- Win-win spiral approach*, yaitu pendekatan dari pendalaman pendekatan *spiral approach*, yang digunakan saat perangkat lunak memiliki tenggat waktu rilis.
- Incremental approach*, yaitu pendekatan yang membagi persyaratan (*requirement*) menjadi beberapa unit fungsional

2.6 Netbeans IDE

Menurut (Fahrin, 2013), Netbeans adalah sebuah *aplikasi Integrated Development Environment* (IDE) yang berbasis *Java* dari *Sun Microsystems* yang berjalan di atas *swing*. *Swing* merupakan sebuah teknologi berbasis *Java* untuk pengembangan aplikasi *desktop* yang dapat berjalan pada berbagai macam platform seperti *windows*, *linux*, *Mac OS X* dan *Solaris*. Sebuah IDE

merupakan lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Graphic User Interface* (GUI), suatu kode editor atau *text*, suatu *compiler* dan suatu *debugger*.

2.7 Swing

Menuru (Steven, 2008) *Swing* adalah *framework* yang berisi *class – class* atau infrastruktur yang dibutuhkan yang digunakan untuk pengembangan aplikasi desktop.

2.8 SQL Server Management Studio

Menurut (Ilman & Fahmi, 2016), umumnya *SQL Server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakan *SQL Server* pada basis data berskala besar. *Microsoft SQL Server* dan *Sybase/ASE* dapat berkomunikasi melalui jaringan yang menggunakan protokol *TDS (Tabular Data Stream)*. *Microsoft SQL Server* juga mendukung *ODBC (Open Database Connectivity)*, dan mempunyai *driver JDBC (Java Database Connectivity)* untuk pemrograman *Java*. Selain itu *Microsoft SQL Server* juga memiliki kemampuan untuk basis data *mirroring* dan *clustering*.

2.9 iReport

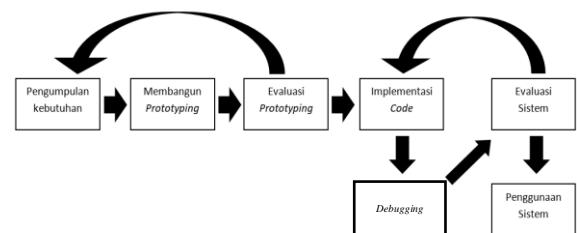
iReport adalah utilitas (perkakas) pelaporan, yang dikembangkan di lingkungan *Java* guna membantu user dan *developer* mendesain laporan secara visual. Menurut (Gata & Gata, 2013), dalam penggunaannya, *JasperReport* atau *iReport* seringkali digunakan oleh pengembang aplikasi berbasis *java*, tetapi banyak sekali pengembang *web* seperti pengembang yang menggunakan bahasa pemrograman *PHP* turut menggunakannya dalam pembuatan laporan. Melalui antar muka yang sederhana namun kaya akan fitur, *iReport* menyediakan fungsi yang paling penting untuk membuat laporan yang kompleks dengan mudah, sehingga akan menghemat banyak waktu. *iReport* menggunakan *library JasperReports* untuk membuat laporan. *JasperReports*, dalam hal ini adalah inti dari *iReport*. *JasperReports* adalah utilitas pelaporan *open source* yang paling populer didalam teknologi *Java*, dan *iReport* adalah *report designer* visual untuk *JasperReports*.

2.10 Visio

Menurut (Lin, Chen, Sun, & Dromey, 2007) dalam bukunya yang berjudul “*Leveraging Single-User Microsoft Visio for Multi-user Real-Time Collaboration*” *Microsoft Visio* adalah salah satu sistem pengeditan grafik *single-user* komersial yang paling lazim, yang dapat digunakan untuk membuat berbagai macam gambar bisnis dan teknis. Salah satu fitur yang membedakan *Visio* dari sistem pengeditan grafik lainnya adalah bahwa rumus, tipe *constraint*, didefinisikan dalam *Visio* untuk mengekspresikan atribut dari setiap objek grafis, dan hubungan antara objek grafis *Visio* yang berbeda.

III. METODOLOGI PENELITIAN

3.1 Metodologi Penelitian



Gambar 1. Prototyping

Ada beberapa tahapan-tahapan dalam metode *prototyping*. Tahapan-tahapan metode *prototyping* seperti yang terlihat di gambar 3.1. Penjelasan gambar 1 adalah sebagai berikut :

1. Pengumpulan Kebutuhan

Tahapan ini adalah tahap awal dimana pengembang dan *user* bersama-sama mendefinisikan format perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun Prototyping

Proses ini adalah perancangan sistem (seperti perancangan *database* dan *use case*) dan design program sementara yang berfokus pada penyajian (contohnya adalah tampilan).

3. Evaluasi Prototyping

Evaluasi ini dilakukan oleh *user* apakah *prototyping* yang sudah dibuat sudah sesuai dengan kebutuhan dan permintaan. Jika sudah sesuai maka dilanjutkan pada langkah selanjutnya. Jika belum sesuai maka direvisi dengan mengulang pada langkah 1, 2, dan 3.

4. Implementasi *Code*

Setelah disepakati *prototyping* yang sudah dibuat maka kemudian diterjemahkan kedalam bahasa pemrograman.

5. *Debugging*

Saat aplikasi sudah selesai, untuk menjadi layak dipakai maka terlebih dahulu harus diuji. Pengujian ini dilakukan agar aplikasi terhindar dari *error* atau *bug*.

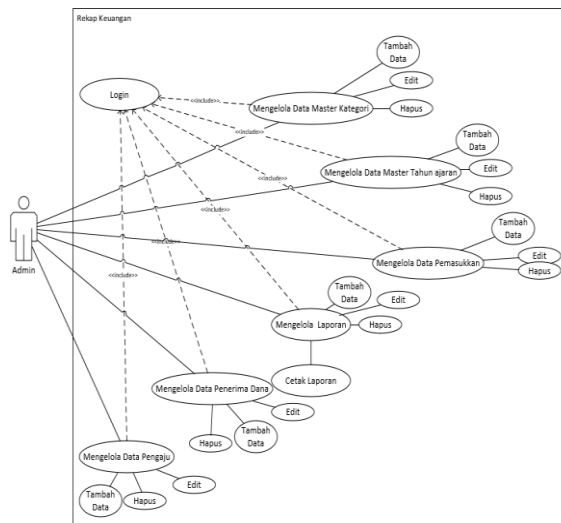
6. Evaluasi Sistem

User akan mengevaluasi apakah sistem yang sudah selesai dibangun sesuai dengan yang diharapkan. Jika iya lanjut ke langkah selanjutnya, namun jika tidak harus mengulangi pada langkah 4 dan 5.

7. Menggunakan Sistem

Aplikasi yang telah diuji dan disetujui oleh *user* maka aplikasi sudah siap untuk digunakan.

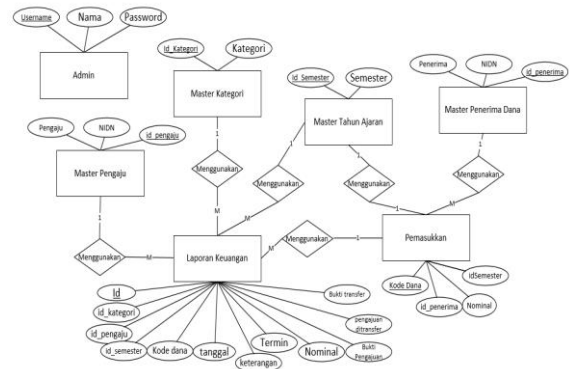
3.2 Use Case Diagram



Gambar 2. Use Case Diagram

Gambar 2 diatas menunjukkan hubungan antara *actor* dan *use case*. Pada gambar diatas menjelaskan pada *aplikasi desktop*, hanya admin yang bisa mengakses aplikasi ini. Admin memiliki hak akses penuh untuk mengolah semua data yang ada, baik tambah data master kategori, tambah data master tahun ajaran, maupun input laporan rekapitulasi dan lain - lain. Dalam kegiatannya, pengelola harus terlebih dahulu melakukan *login*.

3.3 Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram (ERD)

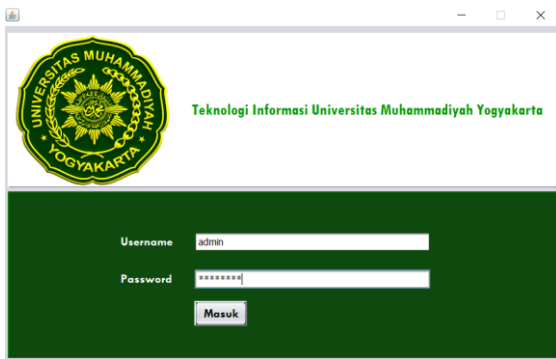
Gambar 3 diatas adalah *ER Diagram* yang menunjukkan entitas dengan atribut dan relasinya. Entitas ini yang nantinya akan menjadi rancangan tabel pada *database* program. Entitas diatas memiliki relasi dari satu dengan yang lain. Beberapa relasi akan dijelaskan sebagai berikut:

1. *Master Kategori* memiliki relasi *one-to-many* dengan *Rekapitulasi*. Satu data kategori dapat digunakan untuk membuat banyak rincian pengeluaran.
2. *Master Tahun Ajaran* memiliki relasi *one-to-many* dengan *Laporan Keuangan*. Satu data tahun ajaran dapat digunakan untuk membuat banyak rincian pengeluaran.
3. *Pemasukkan* memiliki relasi *one-to-many* dengan *Laporan Keuangan*. Satu data pemasukkan dapat dirinci lagi akan digunakan untuk keperluan program studi.
4. *Master Pengaju* memiliki relasi *one-to-many* dengan *Laporan Keuangan*. Satu pengaju dapat mengajukan banyak pengajuan. Pengajuan tersebut akan dicatat dalam rincian pengeluaran.
5. *Master Tahun Ajaran* memiliki relasi *one-to-one* dengan *Pemasukkan*. Satu data data tahun ajaran hanya bisa digunakan untuk membuat satu data pemasukkan.
6. *Master Penerima Dana* memiliki relasi *one-to-many* dengan *Pemasukkan*. Satu

penerima dana bisa
mendapatkan banyak
pemasukkan.

IV. HASIL DAN PEMBAHASAN

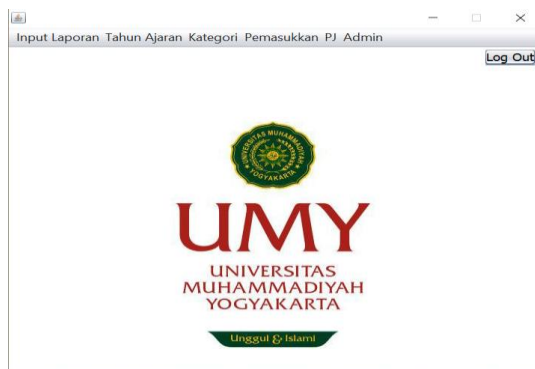
4.1 Antarmuka Login



Gambar 4. Antarmuka Login

Antarmuka *login* adalah halaman dimana admin dapat melakukan *login* dan. Antarmuka *login* dapat dilihat pada gambar 4.

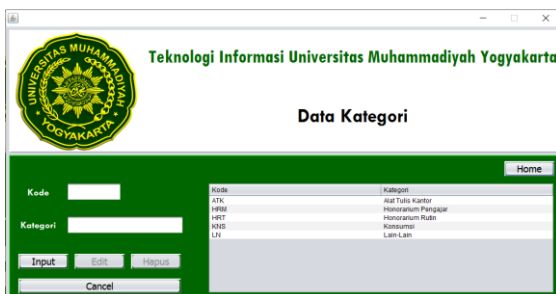
4.2 Antarmuka Home



Gambar 5. Antarmuka Home

Gambar 5 menunjukkan halaman *home*. Setelah login maka aplikasi akan menampilkan halaman *home*. Di halaman ini admin dapat membuka halaman tambah data master kategori, halaman tambah data master tahun ajaran, input laporan, dll.

4.3 Antarmuka Data Master Kategori



Gambar 6. Antarmuka Data Master Kategori

Gambar 6 menunjukkan halaman tambah data master kategori. Halaman ini adalah halaman dimana admin dapat menambah data kategori yang nantinya digunakan dalam pembuatan laporan rekapitulasi.

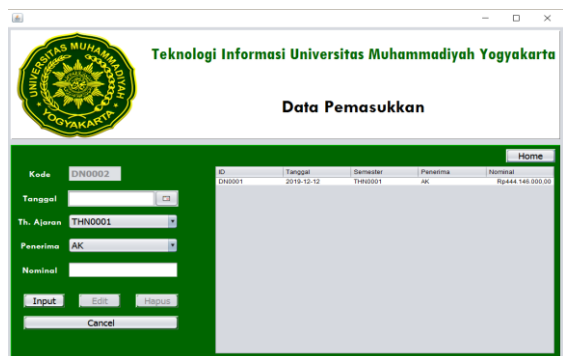
4.4 Antarmuka Data Master Tahun Ajaran



Gambar 7. Antarmuka Data Master Tahun Ajaran

Gambar 7 menunjukkan halaman data master tahun ajaran. Sama dengan dengan halaman tambah data master kategori, admin dapat menambahkan data tahun ajaran yang nantinya akan dipakai saat membuat laporan rekapitulasi.

4.5 Antarmuka Data Pemasukkan



Gambar 8. Antarmuka Data Pemasukkan

Gambar 8 menunjukkan halaman data pemasukkan. Di halaman ini admin dapat menambah, mengubah, dan menghapus data pemasukkan. Di halaman terdapat nominal yang digunakan sebagai saldo untuk tiap data yang di-*input* di halaman input laporan.

4.6 Antarmuka Data Pengaju



Gambar 9. Antarmuka Data Pengaju

Gambar 9 menunjukkan halaman data pengaju. Di halaman ini admin dapat menambah, mengubah, serta menghapus data pengaju. Pada data pengaju terdapat kode dosen yang akan digunakan pada halaman rincian pengeluaran sebagai kode penanggung jawab pengajuan.

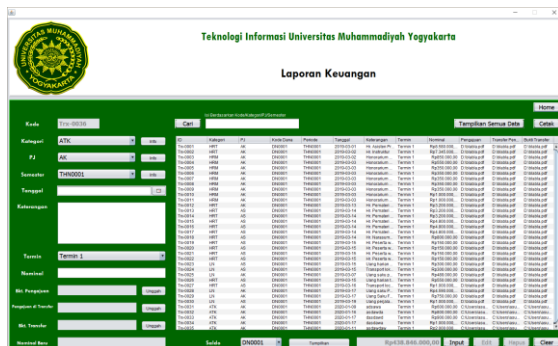
4.7 Antarmuka Data Penerima Dana



Gambar 10. Antarmuka Data Penerima Dana

Gambar 10 menunjukkan halaman data penerima dana. Di halaman ini admin dapat menambah, mengubah, dan menghapus data sesuai dengan kebutuhan. Di halaman ini terdapat kode dosen yang digunakan sebagai kode penerima dana pada halaman memasukkan.

4.8 Antarmuka Laporan Keuangan



Gambar 11. Antarmuka Laporan Keuangan

Gambar 11 menunjukkan halaman laporan keuangan. Halaman ini merupakan fitur utama dari aplikasi ini. Di halaman ini admin dapat melakukan pengolahan data keuangan. Admin juga dapat mencetak laporan. Admin juga dapat melakukan *monitoring* data yang sudah dimasukkan.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian dan pembahasan yang telah dilakukan terhadap *aplikasi desktop* Rekapitulasi Keuangan Program Studi

Teknologi Informasi UMY, maka dapat disimpulkan bahwa :

1. *Aplikasi desktop* dapat digunakan untuk mengelola data keuangan. *Admin* dapat menyimpan, mengubah, dan menghapus data keuangan sesuai dengan kebutuhan.
2. *Admin* dapat melakukan *monitoring* data keuangan dengan menggunakan fitur cari berdasarkan kode transaksi, kategori, tahun ajaran, .
3. *Admin* dapat mencetak rekapitulasi keuangan per-semester.

Dengan demikian *aplikasi desktop* ini sudah dapat memenuhi tujuan dari pengembangan aplikasi.. Setelah tujuan pengembangan tercapai, diharapkan agar *aplikasi desktop* ini dapat bermanfaat bagi Program Studi Teknologi Informasi serta kedepannya masalah – masalah yang ada dapat diselesaikan.

5.2 Saran

Berikut beberapa saran yang dapat digunakan dalam penelitian selanjutnya agar aplikasi ini lebih baik lagi :

1. Menambahkan *form* tanda tangan petinggi program studi di hasil cetak laporan karena untuk verifikasi lebih lanjut perlu adanya tanda tangan tersebut.
2. Rekapitulasi dapat dikelompokkan per-kategori.
3. Menambah fitur untuk menampung data yang telah dihapus seperti *recycle bin*.

DAFTAR PUSTAKA

Chrisianti J., M., & Wijaya, L. (2011). *Aplikasi Pembelian, Penjualan Obat dan Laporan Keuangan Dengan Menerapkan Manajemen Persediaan*. 15.

Fahrudin. (2013). Pengertian Netbeans [Blog]. Diambil dari <http://fahrudin.blogspot.com/2013/05/pengertian-netbeans.html>

Gata, W., & Gata, G. (2013). *Penerapan Bahasa Pemrograman Java Dalam Sistem Informasi Penjualan Versi Desktop*. 82.

- Huda, S. (2012). *SISTEM INFORMASI KEUANGAN BERBASIS DESKTOP DENGAN JAVA STANDARD EDITION & MySQL DI SEKOLAH TINGGI TEKNOLOGI NURUL JADID PAITON PROBOLINGGO*. 26.
- Ilman, & Fahmi, W. (2016). PENGEMBANGAN SISTEM INFORMASI MANAJEMEN TKA/TPA BERBASIS MVC (Studi Kasus di TKA/TPA Al-Ilman Kalibayem) [Blog]. Diambil dari <http://repository.umy.ac.id/handle/123456789/4876>
- Kasmir. (2008). Analisis Laporan Keuangan. Jakarta. (Jakarta).
- Lin, K., Chen, D., Sun, C., & Dromey, G. (2007). Leveraging Single-User Microsoft Visio for Multi-user Real-Time Collaboration.
- Dalam Y. Luo (Ed.), *Cooperative Design, Visualization, and Engineering* (Vol. 4674, hlm. 353–360). https://doi.org/10.1007/978-3-540-74780-2_48
- Lisnawanty, L., Khaldun, I., & Irmayani, W. (2018). APLIKASI LAPORAN KEUANGAN DINAS PENDAPATAN PENGELOLAAN KEUANGAN DAN ASET DAERAH (DPPKAD) KABUPATEN PONTIANAK. *Jurnal Teknik Informatika Musirawas (JUTIM)*, 3(1), 50. <https://doi.org/10.32767/jutim.v3i1.306>.
- Mudakir, R. (2013). *Pengenalan Pengembangan Aplikasi Desktop*.
- Musarofah, S. (2011). *APLIKASI CATATAN KEUANGAN PENJUALAN MEUBEL BERBASIS ANDROID DI TOKO SINAR BARU MAGELANG*. 12(4), 4.
- Permana, F. (2017). *SISTEM INFORMASI BMT INSAN MADANI DENGAN METODE MODEL-VIEW-CONTROLLER*. 147.
- Rusmana, N. Y. (2013). *Sistem Informasi Pengelolaan Keuangan Dana Bantuan Pada Kecamatan Arjosari*. 4.
- Rusmayanti, A. (2014). *Sistem Informasi Pengelolaan Keuangan Pada Desa Ngadirejan*. 6(2), 5.
- Saleh, T. (2017). *SISTEM INFORMASI ADMINISTRASI KEUANGAN MA IBRAHIMY SECANG KALIPURO BANYUWANGI*. 3(1), 8.
- Sri, D., & Wahono, R. S. (2003). Pengantar Unified Modeling Language (UML). Dalam *Ilmu Komputer* (hlm. 1–13).
- Steven, H. (2008). Pengenalan Swing Framework dan Beans Binding di Netbeans IDE [Wordpress]. Diambil dari <https://hendrosteven.wordpress.com/2008/10/16/pengenalan-swing-framework-dan-beans-binding-netbeans-ide/>
- Wulansari, N., Purnama, B. E., & Wardati, I. U. (2013). *Sistem Informasi Pengelolaan Keuangan Sekolah Pada Sekolah Menengah Kejuruan (SMK) Pgri 1 Pacitan*. 7.
- Zihni, F., & Hius, J. J. (2015). *RANCANG BANGUN APLIKASI PEREKAPAN LAPORAN KEUANGAN PADA STUDIO MULTIMEDIA NADSTARR ENTERTAINMENT*. 9.

PENULIS:

Tri Yoga Utama

Teknik Informatika, Fakultas Teknik,
Universitas Muhammadiyah Yogyakarta,
Yogyakarta.

Email: tri.yoga1996@gmail.com