

PERANCANAGAN APLIKASI ACCOUNT PAYABLE SUBSIDIARY DENGAN PEDEKATAN MODEL PHASED DEVELOPMENT

Edy Nasri¹, Widyawati², Sodikin³

^{1,2}Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Banten Jaya,
Jl. Syeh Nawawi Albantani, Curug, Serang – Banten

e-mail: ¹edynasri@unbaja.ac.id, ²widyawati@unbaja.ac.id, ³Sodikina963@gmail.com

Abstract

In this era of technological development, a lot of stored information requires applications that can support all activities, effectively and efficiently, but not many companies have implemented the application as a whole. one of which is Amaris Hotel Cilegon which has not used applications for recording purchases, receipts and controlling debts. Debt data processing still uses spreadsheets which are prone to various problems. Therefore, this study aims to analyze work activities by looking at the system that is running at Amaris hotel Cilegon. The data on this study was collected using literature study methods and field studies to get accurate data results. Application creation uses native PHP language programming, with MySQL database and Class Bootstrap as a framework interface, while in the design of application system development the method used is the RAD method with Phased Development Model. In this study, the author designed and developed the Account Payable Subsidiary application that can provide the company's debt control features at any time, record-keeping relationships with payable accounts and performed only one process in the recording process receiving, to produce output in the form of values listed in receiving journals and payable account journals and can be directly used as the basis for making financial statements by Bookkeeper Accounting Supervisor.

Keyword: Account Payable, Phased Model, RAD, Subsidiary

PENDAHULUAN

Kemampuan teknologi informasi masa sekarang telah membantu dalam menciptakan suatu aplikasi yang handal, kompeten dan akurat, salah satunya aplikasi akuntansi, Namun belum banyak perusahaan yang menerapkan aplikasi akuntansi terhadap siklus pendapatan dan pengeluaran guna mengatasi ancaman – ancaman yang dihadapi oleh perusahaan. Salah satunya ancaman pencatatan laporan pembelian yang belum relevan dan pembayaran utang yang tidak terkontrol, maupun hilangnya data serta kesalahan atau redudansi data sangat memungkinkan untuk terjadi dan akan menjadi masalah bagi perusahaan yang belum menerapkan sistem informasi akuntansi.

Suatu perusahaan harus memiliki sistem informasi yang terintegrasi yang dapat mencakup kegiatan tertentu, salah satunya adalah sistem informasi akuntansi(Lusiansyah, 2022). Pada perusahaan ini dalam melakukan pencatatan hutang dagang masih dilakukan dengan menggunakan Spreadsheet(Hanif, 2019). Pencatatan tersebut masih memiliki banyak kekurangan dimana Menu yang tersedia tampak tidak beraturan sehingga masih sulit untuk membuat fitur kontrol hutang dagang tersebut, susah untuk mensikronkan data yang ada pada perusahaan dan mudah terjadi redudansi data. Secara rinci permasalahan yang terjadi yaitu dimulai dari divisi Purchasing, proses pembuatan form pemesanan barang (*purchase order*) yang dilakukan menggunakan spreadsheet membuat tidak adanya efisiensi dan efektifitas kerja dengan pekerjaan pencatatan penerimaan barang pada divisi *Receiving*, sehingga ketika ada barang dari *purchase order* datang, Divisi *Receiving* harus mencatat ulang barang-barang apa saja dan berapa banyak yang datang/diterima , selanjutnya di pencatatan transaksi penerimaan barang tersebut dan pencatatan beban hutang (Divisi *Account Payable*) juga menghilangkan efisiensi dan efektifitas kerja, proses nya harus dilakukan satu persatu, sehingga sering kali

terjadi kesalahan pencatatan antara kedua divisi tersebut, yang sebetulnya pencatatan tersebut bisa dilakukan hanya sekali pada divisi Receiving dan akan secara otomatis masuk menjadi catatan beban hutang yang masuk ke dalam pencatatan *Account Payable*, permasalahan lain nya yaitu dalam proses pembuatan jurnal dari hasil pencatatan divisi purchasing, *Receiving* , *Account payable* dan satu lagi yaitu *Inventory Store* masih dilakukan dengan sama sehingga sering terjadi kesalahan penjurnalan dan mengakibatkan tidak balance nya pada kas biaya keluar.

Selain itu perusahaan juga memiliki masalah lain pada siklus pengeluaran ini, yaitu belum terdapat pengontrolan terhadap laporan utang perusahaan, karena hal tersebut perusahaan tidak dapat melihat laporan hutang yang sudah dibayarkan maupun belum dibayarkan.

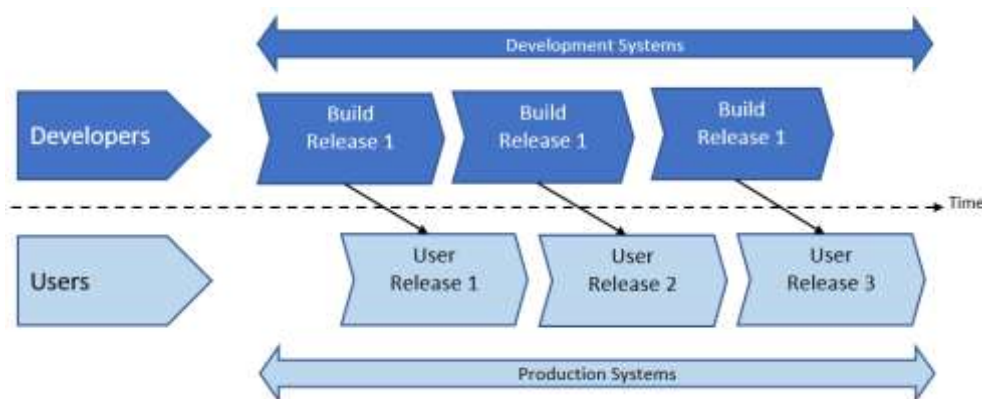
Dari uraian di atas, maka perusahaan membutuhkan aplikasi yang membantu untuk memecahkan permasalahan tersebut, dapat menangani pengelolaan beban secara terperinci. Sehingga saat melakukan transaksi, data tersebut dapat mengenerate langsung ke jurnal. Dalam pembuatan aplikasi ini digunakannya Bahasa PHP (Hypertext Preprocessor) dimana Bahasa tersebut dirancang untuk membentuk halaman Web yang dinamis dan dalam penyimpanan seluruh data aplikasi ini disimpan pada DBMS MySql (Riyansyah, 2021). Aplikasi ini akan memuat fitur-fitur siklus pengelolaan data pekerjaan yang saling berhubungan antara divisi Purchasing, Receiving, Store Inventory, dan Account Payable, untuk hasil output aplikasi ini yaitu berupa jurnal dari masing-masing departemen tersebut yang selanjutnya jurnal tersebut akan di jadikan sebagai salah satu dasar oleh Seorang *Bookeper* untuk membuat laporan keuangan pada bagian pengeluaran biaya (*Expenses*) (Alazmi et al., 2021; Diaz Arizona & Susilowati, 2021; Hidayah et al., 2020).

METODE PENELITIAN

Bagian ini menjelaskan secara rinci tentang penelitian yang dilakukan. Berikut ini merupakan tahapan dalam melakukan penelitian yang dilakukan:

1. Metode Pengembangan Sistem

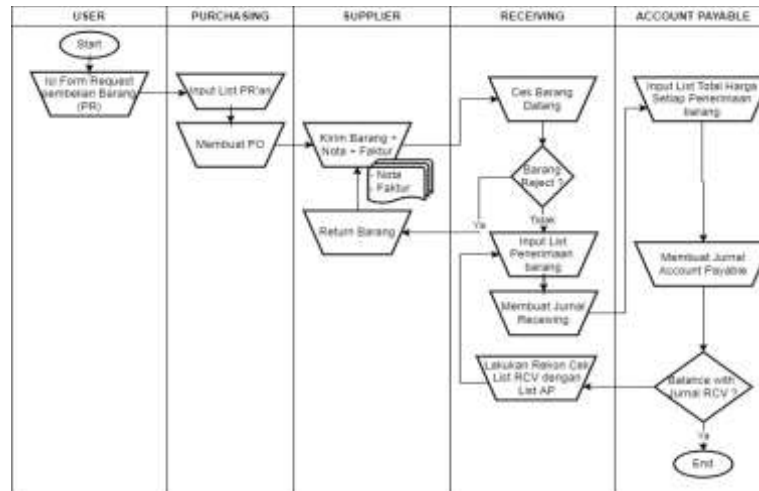
Rapid Application Development (RAD) merupakan sebuah metode pengembangan perangkat lunak yang bersifat customize yang adaptasi kecepatan tinggi dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen (Ohta et al., 2005).



Gambar 1. Model Phased Development

2. Diagram Alir pada rancangan sistem

Meggambarkan dokumen yang mengalir dan proses yang dilakukan



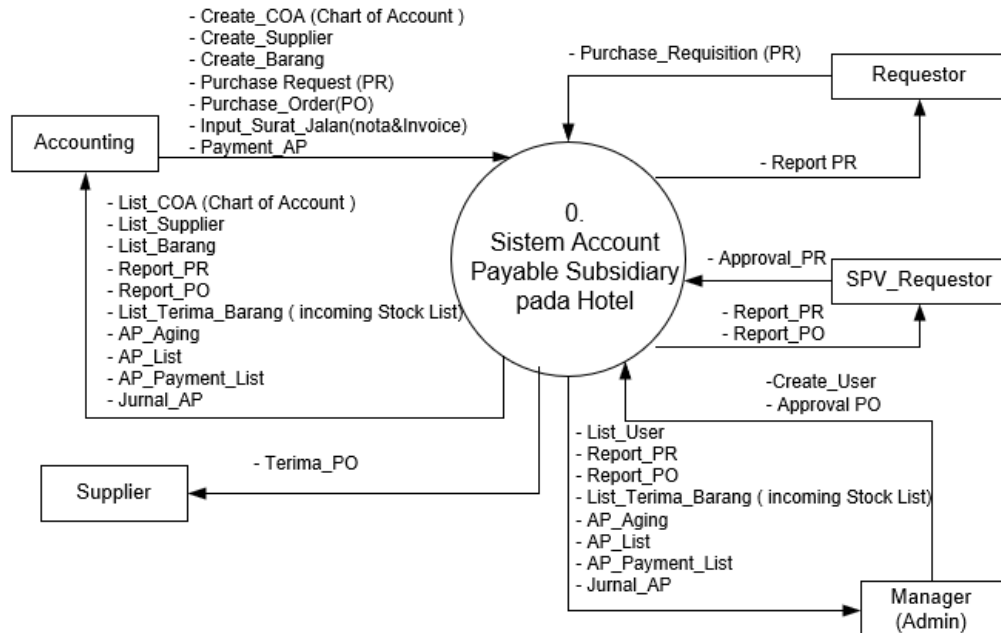
Gambar 2. Perancangan Diagram Alir Sistem

3. Perancangan diagram arus data

Dalam perancangan ini akan dimodelkan dengan proses alur data sistem untuk Sistem Account Payable Subsidiary dengan membuat diagram context (Inti & Issn, 2021) dan diagram berjenjang, diagram alir data Context (Muliadi et al., 2020) atau level 0, DFD Overview atau level 1 dan DFD Rinci atau level 2.

1.1. Context Diagram Rancangan

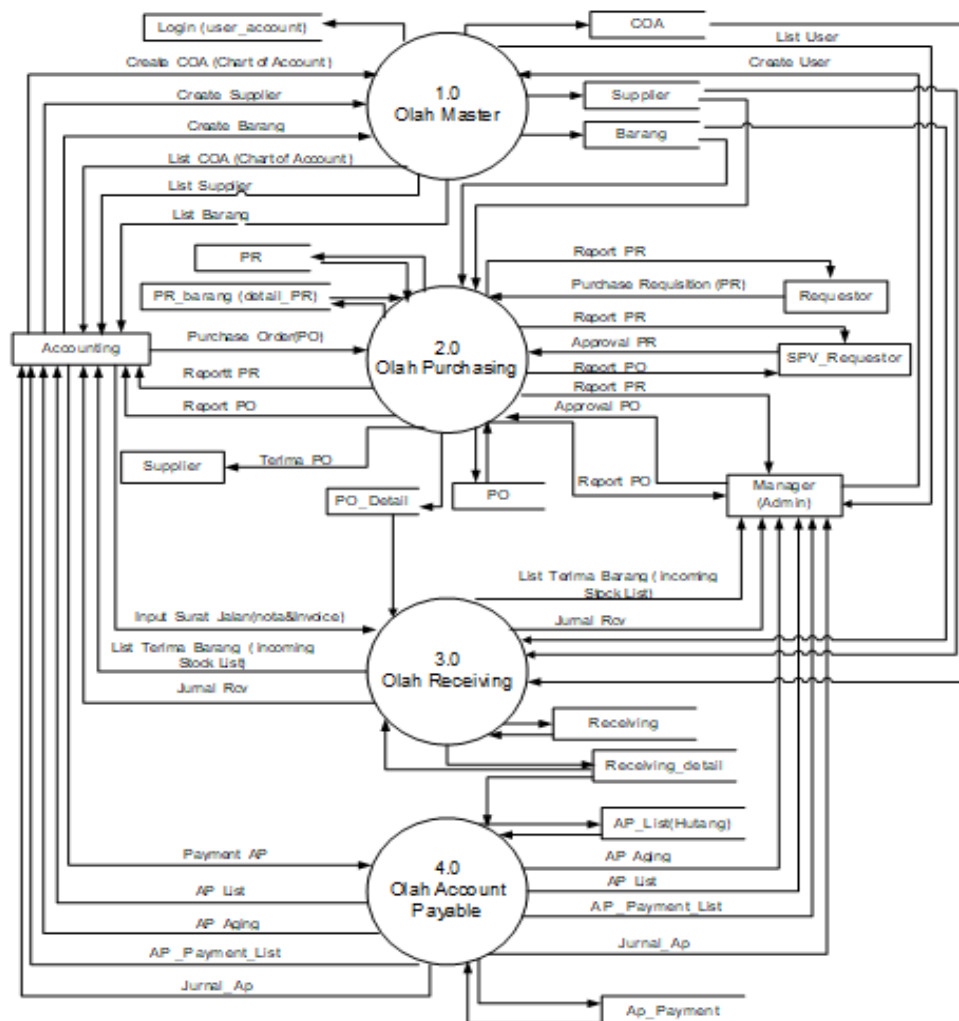
Berikut ini pemodelan *Context Diagram* sistem *Account Payable Subsidiary*



Gambar 3. Context Diagram Rancangan Sistem

1.2. Overview Diagram Rancangan

Berikut ini pemodelan *Overview Diagram* sistem *Account Payable Subsidiary*

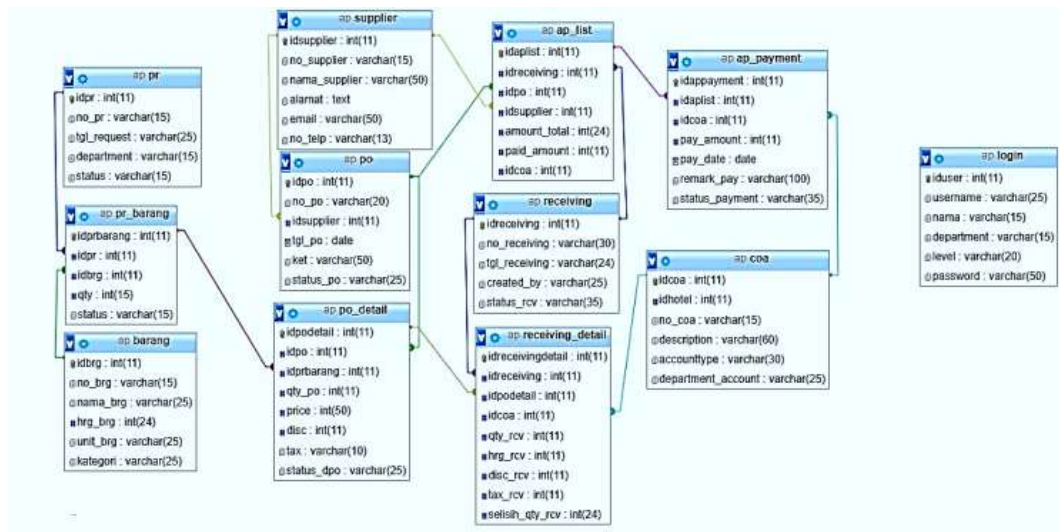


Gambar 4. Overview Diagram Rancangan Sistem

4. Rancangan Basis Data

Entity Relationship Diagram (ERD), digunakan untuk menggambarkan representasi grafis, dimana langkah-langkahnya melibatkan: (1) Mengidentifikasi, menganalisis, dan menyempurnakan aturan bisnis yang melibatkan mempelajari bisnis kegiatan, mengumpulkan informasi dan menemukan elemen data dengan mengidentifikasi informasi, kebutuhan, pengguna, sumber dan konstitusi, (2) Identifikasi entitas utama, (3) Tentukan hubungan antar entitas, (4) Setiap langkah dijelaskan dan diagram untuk menggambarkan proses (Amran N, et al., 2018).

Menurut Rahmayu dalam jurnal (Tabrani & Priyandaru, 2021) menjelaskan mengenai ERD merupakan diagram yang menunjukkan informasi yang dibuat, disimpan dan digunakan dalam sistem bisnis. Entity Relation Diagram digunakan untuk basis data relasional. Berikut gambar *Entity Relation Diagram* (Simanjuntak et al., 2017) dan *ERD Physica* (Faradhila et al., 2019).



Gambar 5. *ERD Physical Aplikasi*

HASIL DAN PEMBAHASAN

A. Implementasi Sistem

1. Autorisasi

Untuk akses (masuk atau *login*) aplikasi dibutuhkan *username* dan *password* jika berhasil masuk maka akan tampilan halaman dashboard sesuai dengan otoritas yang diberikan:

a) Halaman Dashboard Admin (Hotel Manager)

Level Admin akan dapat mengakses banyak menu-menu: (1) menu *master create user*, (2) menu *purchasing* yaitu melihat *report purchase request*, *approval / edit PO* dan melihat *report PO*, (3) menu *receiving* dapat melihat *incoming list* dan *report Jurnal RCV*, (4) menu *account payable* dapat melihat *a/p list*, *a/p aging list*, *a/p payment list* dan *report jurnal a/p*.

b) Halaman *Dashboard Accounting*

Level *Accounting* dapat mengakses menu-menu diantaranya: (1) menu master, yaitu melakukan olah barang, *Supplier* dan *Coa*, dalam hal ini yaitu *create*, *edit* dan *delete* (2) Menu *Purchasing*, yaitu *create pr* disertai dengan *edit* dan *delete pr* serta *report pr* dapat juga dilihat, selain itu dalam menu *purchasing* dapat melakukan *create purchase Order* serta melihat *report po* yang telah di buat. (3) Menu *receiving*, yaitu proses *create incoming item*, melihat *list incoming item*, *create jurnal rcv* serta melihat *report jurnal rcv* yang telah dibuat. (4) Menu *Account payable*, yaitu melihat *a/p list*, melihat *a/p aging list*, proses *payment*, melihat *payment list*, *create Jurnal a/p* serta melihat *report jurnal a/p* yang telah dibuat.

c) Halaman *Dashboard Supervisor*

Level *Supervisor* dapat mengakses menu-menu berikut, yaitu : (1) Menu *Purchasing*, pada level *supervisor* dapat melakukan *create pr*, *edit/approval pr*, *delete pr* dan juga melihat *report pr* yang telah dibuat, Serta *supervisor* dapat melihat *report purchase order* yang telah dibuat oleh *accounting*.

d) Halaman *Dashboard Requestor*

Level *requestor* diberikan hak akses dalam menu purchasing yaitu untuk *create pr*, serta melihat *report PR* yang telah dibuat

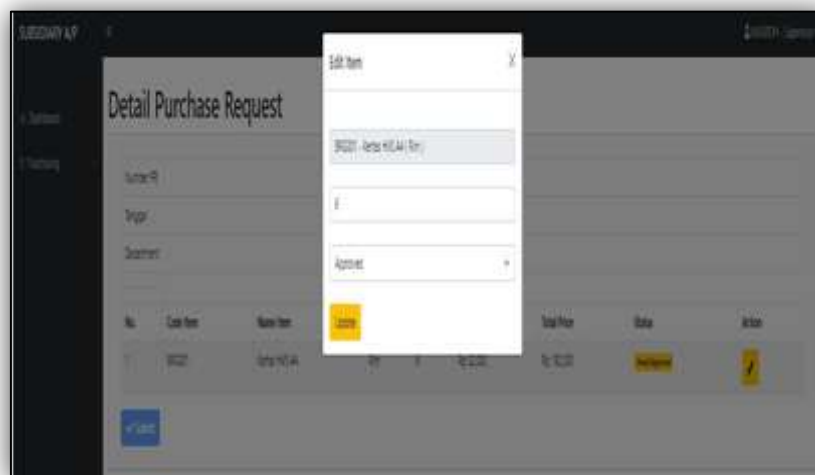
2. Dalam aplikasi ini, ada menu Data master yang ada pada level *accounting*, untuk menambah, mengedit, dan menghapus data-data diantaranya yaitu data barang, data *supplier* dan data *CoA (Chart of Account)*.

3. Pada akses level manager (Admin), memiliki akses data *master user*, digunakan untuk menambah , mengedit serta menghapus data *user*
4. Di dalam level requestor, user diberikan akses menu *purchasing*, yaitu: *Purchase Request* serta *Report Purchase Request*. Menu *Purchase Request* diperuntukan untuk membuat permintaan pembelian barang yang dibutuhkan kepada *department Purchasing – Accounting*, sedangkan menu *Report Purchase Request* untuk melihat status dari PR yang sudah dibuat, apakah sudah di *approved* oleh *supervisor* masing-department yg membuat PR, atau belum. Berikut tampilan halaman *Purchase Request* dan *report PR*.



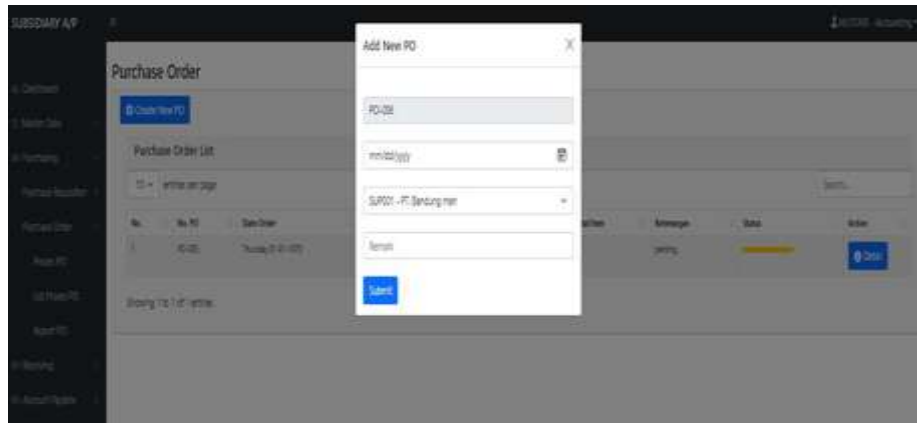
Gambar 6. Tampilan Halaman Report PR

5. Setelah *create PR* sudah dilakukan, data PR tersebut akan ditampilkan di halaman menu list proses PR pada *user level Supervisor* untuk bisa dilakukan *Edit* dan *Approve PR*, seperti pada gambar berikut:



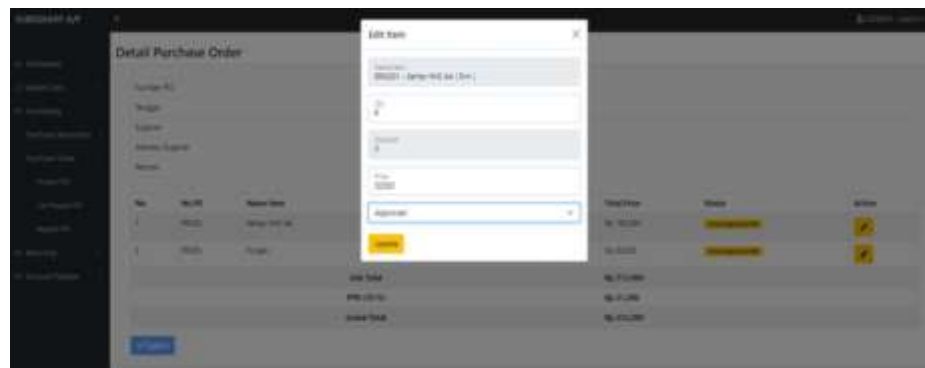
Gambar 7. Tampilan Halaman Proses Edit / Approval PR

6. Setelah proses PR sudah selesai, maka *user* dengan *level accounting* bisa melihat PR *approved* pada menu yang sama yaitu *Report PR Approved* dan menarik data PR tersebut untuk dilakukan pemesanan (*Purchase Order*) kepada *supplier* yang sudah ditentukan, proses pemuatan PO diantaranya yaitu seperti pada gambar berikut:



Gambar 8. Tampilan Halaman Create PO

8. Selanjutnya, PO tersebut akan di tampilkan di halaman menu *List Proses PO* pada level Manager (Admin) untuk dilakukan proses *Approvall* PO sebelum PO tersebut bisa di cetak dan dikirimkan ke *supplier*, setelah selesai klik *button submit* untuk memproses *Approve* PO, seperti pada gambar di bawah:



Gambar 9. Tampilan Halaman Proses Edit/Approval PO

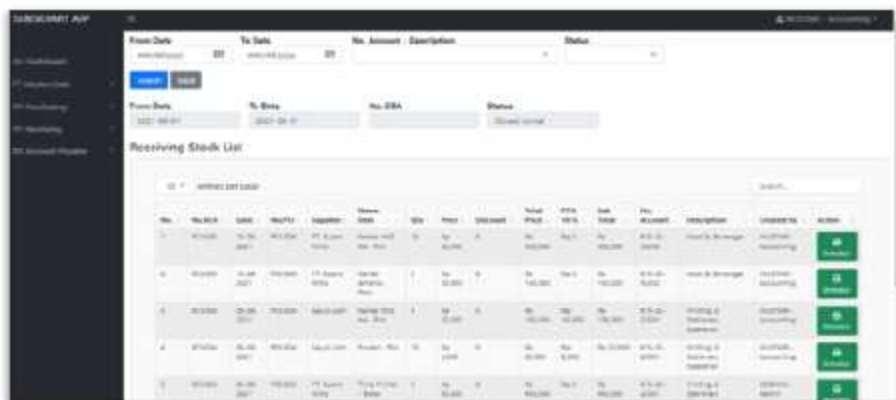
9. Selanjutnya Ketika barang PO sudah dikirimkan oleh *supplier*, lakukan proses penerimaan barang PO tersebut, setelah itu lakukan pencatatan penerimaannya sesuai dengan nota / surat jalan, langkah nya yaitu klik *button Create New Incoming*, isi tanggal penerimaan barang dan Klik *Submit*, seperti pada gambar berikut:



Gambar 10. Tampilan Halaman Proses Receiving Incoming by PO

10. Setelah itu, klik *button detail* pada *receiving incoming by po* yang telah di buat, tampilan akan berpindah ke halaman detail *incoming by po*, selanjutnya klik *add item incoming by po* untuk memilih item-item pada PO yang akan di proses *incoming*, selanjutnya klik *button submit* untuk memproses *incoming by po* tersebut.

11. Selanjutnya *incoming by po* tersebut akan ditampilkan pada menu *Incoming Item List*, seperti pada gambar berikut:



Gambar 11. Tampilan Halaman *Incoming Item List*

12. Di akhir bulan, accounting akan *create* jurnal dari dengan *from date* nya tanggal 1 dan *to date* nya tanggal akhir pada bulan berjalan, untuk *close* penerimaan pada bulan tersebut, hasil *create* jurnal bisa dilihat pada *report jurnal rcv*.
13. Setelah melakukan *Incoming Item*, semua daftar penerimaan barang yang ada dalam List *Incoming Item* akan ditampilkan juga pada halaman menu *Account Payable* → *A/P List* dengan format tampilan table yang berbeda sebagai List Hutang perusahaan, dan juga akan terakumulasi dan ditampilkan ke halaman *Account Payable* → *A/P Aging List* untuk mengetahui sudah memasuki umur berapa bulan hutang tersebut.
14. Setelah mengetahui jumlah hutang kepada supplier-supplier, dan sudah dilakukan proses pembayaran melalui bank terhadap hutang tersebut, Pada menu *Account Payable* → *Payment A/P*, lakukan pencatatan pembayaran tersebut agar mengurangi daftar hutang pada *A/P Aging List* dan *A/P List*. Pencatatan *Payment* tersebut akan di tampilkan di halaman *A/P Payment List*, di akhir bulan, setelah semua pencatatan penerimaan barang sudah di lakukan, buat jurnal *A/P* pada menu *Account Payable* → *A/P jurnal*, *A/P Jurnal* ini berisikan akumulasi total Hutang dengan debit nya sebagai *Cost / Expense* dan *Credit* sebagai Total Hutang *A/P (A/P Trade)*.

B. Uji Coba

Dalam tahapan pengembangan aplikasi diperlukannya Tapan Uji Coba atau testing dalam didalam melakukan testing black box adalah mengidentifikasi permasalahan dengan cara memasukan data uji kedalam aplikasi (Nasri et al., 2022). Tabel hasil pengujian aplikasi yang dilakukan dalam metode *black box* sebagai berikut:

Tabel 1. Pengelompokan pengujian *blackbox*

No	Item Uji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
1	Menu <i>Account Payable</i>	<i>A/P List</i>	Berhasil menampilkan daftar Hutang <i>A/P</i> sesuai dengan <i>range</i> tanggal dan supplier yang dipilih	Valid
		<i>A/P Aging List</i>	Berhasil menampilkan total hutang per-supplier sesuai dengan umur hutangnya	Valid
		<i>A/P Payment List</i>	Berhasil menampilkan daftar <i>payment</i> sesuai dengan <i>range</i> tanggal dan status <i>list payment</i>	Valid

		Report Jurnal A/P	Berhasil menampilkan daftar <i>report jurnal A/P</i> yang berstatus 'Closed Jurnal'	Valid
2	Menu Master Data (COA)	Coa	Berhasil menampilkan halaman <i>list coa</i> yang sudah terdaftar	Valid
		Add New Coa	Berhasil menambahkan COA	Valid
		Edit	Berhasil mengubah data COA	Valid
		Delete	Berhasil menghapus COA	Valid
3	Purchase Requisition	Proses PR	Berhasil menampilkan PR yang sudah dibuat dan membutuhkan <i>approval supervisor</i>	Valid
		Add New PR	Berhasil membuat PR No. PR baru	Valid
		Detail	Berhasil menampilkan halaman detail PR dan menambahkan item-item ke dalam PR yang di pilih	Valid
		Delete	Berhasil Menghapus No. PR	Valid
		PR List Approved	Berhasil menampilkan daftar No. PR yang sudah di <i>Approve</i>	Valid
		PR List Rejected	Berhasil menampilkan daftar No. PR yang di <i>Reject</i>	Valid
4	Purchase Order	Proses PO	Berhasil menampilkan halaman untuk membuat PO	Valid
		Add New PO	Berhasil membuat PO No. PO baru	Valid
		Detail	Berhasil menampilkan halaman detail PO dan menambahkan item-item ke dalam PO yang di pilih	Valid
		Delete	Berhasil Menghapus No. PO	Valid
		List Proses PO	Berhasil menampilkan <i>list PO</i> yang membutuhkan <i>Approval</i> .	Valid
		Report Po	Menampilkan halaman <i>list report po</i>	Valid
		Search	Menampilkan <i>list report po</i> sesuai dengan <i>range</i> tanggal dan status po yang dipilih	Valid
		Detail	Berhasil menampilkan <i>item-item PO</i> sesuai dengan no PO yang di pilih	Valid
		Print	Berhasil mencetak PO yang dipilih	Valid
		Back	Berhasil kembali ke halaman sebelumnya	Valid
5	Menu Account Payable	A/P List	Berhasil menampilkan halaman <i>A/P List</i>	Valid
		Search	Berhasil menampilkan daftar Hutang A/P sesuai dengan <i>range</i> tanggal dan supplier yang dipilih	Valid
		A/P Aging List	Berhasil menampilkan total hutang per-supplier sesuai dengan umur hutangnya	Valid
		A/P Payment	Berhasil menampilkan daftar <i>AP/List</i> yang belum di <i>payment</i> untuk di proses <i>payment</i>	Valid
		A/P Payment List	Berhasil menampilkan daftar <i>payment</i> sesuai dengan <i>range</i> tanggal dan status <i>list payment</i>	Valid
		Create A/P Jurnal	Berhasil menampilkan halaman <i>Create Jurnal A/P</i>	Valid
		Search (Jurnal A/P)	Berhasil menampilkan <i>Jurnal A/P List</i> sesuai dengan <i>range</i> tanggal, yang dipilih	Valid
		Submit	Berhasil melakukan <i>Generate Jurnal A/P</i> sesuai dengan <i>range</i> tanggal yang di pilih, dan status <i>A/P payment</i> menjadi "Closed Jurnal"	Valid
		Report Jurnal A/P	Berhasil menampilkan daftar <i>report jurnal A/P</i> yang berstatus 'Closed Jurnal'	Valid
		Back	Berhasil kembali ke halaman sebelumnya	Valid

KESIMPULAN

Berdasarkan penelitian yang dilakukan penulis memperoleh kesimpulan sebagai berikut :

1. Aplikasi Account Payable Subsidiary di buat dengan menggunakan Bahasa Pemrograman PHP Native dengan Framework Bootstrap dan menggunakan DBMS MySQL sebagai *database*.
2. Menyediakan fitur pengontrolan hutang yaitu pada sub menu *A/P Aging List* hutang yang masih *Outstanding*, *aging* berapa bulan hutang tersebut.
3. Mampu melakukan pekerjaan pencatatan *receiving* dengan *account payable* hanya satu kali proses, pencatatan dalam aplikasi ini proses *receiving* dan *account payable* dapat di relasikan, sehingga efisiensi dapat terpenuhi.
4. Mempunyai hasil output berupa nilai-nilai yang tercantum dalam jurnal *receiving* dan jurnal *account payable* dan sudah langsung bisa di jadikan dasar untuk membuat *General Ledger Bookeper* dalam hal *account-account subsidiary Account Payable* dan juga output berupa *purchase order* sebagai media untuk melakukan pemesanan barang

SARAN

Setelah melakukan penelitian dan pengujian aplikasi *subsidiary account payable* yang dikembangkan, maka saran - saran yang mungkin bermanfaat bagi pengembang aplikasi yang akan datang yaitu mengintegrasikan dengan *system accounting*.

DAFTAR PUSTAKA

- Alazmi, Umami, N., Lisnawanty, & Ardiansyah. (2021). Sistem Informasi Akuntansi Pengelolaan Keuangan pada Davia Laundry Kubu Raya. *Jurnal Sistem Informasi Akuntansi (JUSTIAN)*, 2(1), 01–10.
- Amran N, Mohamed H & Diana F (2018). Developing Human Resource Training Management (HRTM) Conceptual Model Using Entity Relationship Diagram (ERD). *International Journal of Academic Research in Business and Social Sciences*, 8(12),1444-1459
- Hidayah, A., Syarif, M., Latifah, & Nurmalasari. (2020). Sistem Informasi Akuntansi Peminjaman Dan Pembayaran Angsuran Berbasis Web Pada Kantor Unit Pengelolaan Kegiatan (Upk) Sejangkung. *Jurnal Sistem Informasi Akuntansi (JUSTIAN)*, 3(1), 01–11.
- Diaz Arizona, N., & Susilowati, E. (2021). Sistem Informasi Akuntansi Simpan Pinjam Pada Sinar Kalimantan Dengan Metode Pengembangan Waterfall. In *Jurnal Sistem Informasi Akuntansi ρ* (Vol. 02, Issue 01).
- Faradhila, A., Setiawan, R., Stekpi, J., Trilogi, /, Kalibata, T., & Selatan, J. (2019). Rancang Bangun Aplikasi Pengolahan Data Merchant Berbasis Web Pada Pt. Finnet Indonesia. In *Jurnal Sistem Informasi dan Sains Teknologi* (Vol. 1, Issue 1). www.mysql.com.
- Hanif, A. (2019). Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) Sistem Informasi Sederhana Menggunakan Spreadsheet dan Macro Untuk Usaha Mikro Informal. *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)*, 851–855. <https://seminar-id.com/semnas-sainteks2019.html>
- Inti, J., & Issn, T. (2021). *Rancang Bangun Aplikasi E-Commerce Kosmetik Menggunakan Payment Gateway Berbasis Web* (Vol. 13, Issue 01).

- Lusiansyah, Y. (2022). The Influence of System User Ability and Top Management Support on Accounting Information System Performance. *JASa (Jurnal Akuntansi, Audit Dan Sistem Informasi Akuntansi)*, 6(1), 148–159. <https://doi.org/10.36555/jasa.v6i1.1843>
- Muliadi, M., Andriani, M., & Irawan, H. (2020). Perancangan Sistem Informasi Pemesanan Kamar Hotel Berbasis Website (Web) Menggunakan Data Flow Diagram (DFD). *JISI: Jurnal Integrasi Sistem Industri*, 7(2), 111. <https://doi.org/10.24853/jisi.7.2.111-122>
- Nasri, E., Kornelis., & Syamsuddin, H. (2022). *Perancangan Aplikasi Warehouse Management Dengan Pendekatan Metode Fuzzy Mamdani*. Jurnal Saintek (Vol. 6, Issue 01).
- Ohta, K., Yoshida, K., Tanabe, K., Hihara, J., Yamaguchi, Y., & Toge, T. (2005). A case report of bowel obstruction due to peritonitis carcinomatosa using combined administration with CPT-11 and CDDP. In *Gan to kagaku ryoho. Cancer & chemotherapy*.
- Riyansyah, A. (2021). Perancangan Sistem Informasi Penggajian Berbasis Web Menggunakan Php Dan Mysql Di Lion Parcel Cisokan Kota Bandung. *Infotech Journal*, 29–35. <https://doi.org/10.31949/infotech.v7i2.1373>
- Simanjuntak, H., Lumbantoruan, R., Banjarnahor, W., Sitorus, E., Panjaitan, M., & Panjaitan, S. (2017). Program Studi Sarjana Sistem Informasi. In *Jl. Sisingamangaraja, Sitoluama* (Vol. 6, Issue 1). <http://www.diffnow.com/>
- Tabrani, M., & Priyandaru, H. (2021). Sistem Informasi Manajemen Berbasis Website Pada Unl Studio Dengan Menggunakan Framework Codeigniter. In *Jurnal Ilmiah M-Progress* (Vol. 11, Issue 1).