

SISTEM INFORMASI INVOICE BERBASIS WEB DALAM PENGAJUAN BIAYA OPERASIONAL PADA PT. “XYZ”

David Winata

Universitas Dirgantara Marsekal Suryadarma

d4vid.winnata@gmail.com

Abstraksi

Sistem pengajuan invoice merupakan system yang menawarkan kemudahan dan kerapihan pada penyimpanan dan pelaporan pengajuan invoice. Sistem ini dibentuk menggunakan bahasa pemrograman java dan PostgreSQL untuk database. Bahasa pemrograman java merupakan bahasa yang paling banyak digunakan oleh berbagai kalangan, selain itu bahasa pemrograman java merupakan bahasa pemrograman yang saya kuasai. Oleh karena itu, saya menggunakan bahasa pemrograman java sebagai dasar untuk pembuatan sistem informasi pengajuan invoice. Sedangkan untuk metode yang digunakan adalah metode waterfall, karena dengan metode ini pengerjaan akan berurutan dan lebih terstruktur. Sistem informasi invoice ini memudahkan karyawan pada PT XYZ dalam pengajuan dan pelaporan invoice. Dengan menggunakan sistem informasi ini karyawan akan lebih mudah dan lebih efisien dalam pengajuan invoice biaya operasional. Kedepannya, system ini diharapkan dapat dibuatkan untuk mobile dan terdapat fitur yang lebih beragam.

Kata Kunci : Invoice, Java, Sistem Informasi, PT XYZ

Abstract

The invoice submission system is a system that offers convenience in submitting invoices and neatness in the storage and reporting of invoice submissions. This system is built using the Java programming language and PostgreSQL for databases. The Java programming language is the language most widely used by various groups, besides the Java programming language is a programming language that I master. Therefore, I use the Java programming language as the basis for creating an invoice submission information system. While, for the method used is the waterfall method, because with this method the work will be consecutive and more structured. This invoice information system makes it easier for employees at PT XYZ to submit and report invoices. By using this information system, employees will find it easier and more efficient in submitting invoices for operational costs. In the future, this system is expected to be available for mobile and there are more diverse features.

Keyword : Invoice, Java, Information System, PT XYZ

PENDAHULUAN

Di era globalisasi saat ini peran teknologi informasi sangat penting untuk memenuhi kebutuhan informasi yang akurat, tepat dan cepat. Perkembangan teknologi informasi telah membawa dampak besar terhadap kemajuan pengolahan data dan penyampaian informasi, salah satu

wujud nyata dari dampak kemajuan teknologi informasi ini dapat dilihat pada sistem informasi operasional di PT.XYZ

PT.XYZ merupakan perusahaan yang menyediakan tenaga kerja mahir dan terpercaya. PT.XYZ mengkhususkan diri dalam menyediakan tenaga

kerja yang terampil dan andal untuk semua sektor industri dan melayani semua tingkat keterampilan.

Sayangnya proses pendataan pada PT.XYZ masih bersifat manual, sehingga membutuhkan waktu yang lama. Proses pengajuan *invoice* merupakan salah satu proses pendataan dan pengajuan yang masih menggunakan metode manual. Hal ini menjadikan sistem pelaporan yang sering menemui kesalahan peng-*input*-an maupun kesalahpahaman terhadap pengajuan *invoice*.

Berdasarkan permasalahan tersebut maka perlu dibuatkan suatu sistem informasi teknis operasional agar proses pengajuan *invoice* lebih cepat, tepat dan efisien pada PT.XYZ Sistem informasi teknis operasional yang akan dirancang berbasis *website* karena penggunaan *website* lebih sederhana untuk digunakan dan penggunaannya lebih mudah dipahami.

Dengan adanya sistem informasi pendataan elektronik ini diharapkan kebutuhan dan masalah yang sering terjadi pada proses pengajuan *invoice* dapat teratasi dengan baik, sehingga dapat meningkatkan kinerja dari pihak-pihak terlibat dan dapat menghasilkan data secara digital dalam pengarsipannya.

Metode Pengumpulan Data

Pengumpulan data untuk penelitian menggunakan pendekatan sebagai berikut :

- a. Observasi
Observasi dilakukan dengan mengamati secara langsung keadaan PT.XYZ untuk pembuatan sistem informasi *invoice* berbasis *online*.
- b. Wawancara

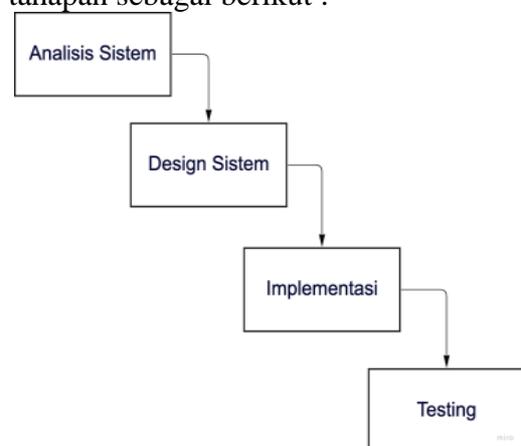
Wawancara dilakukan dengan cara tanya jawab dengan berbagai karyawan yang meliputi admin operasional dan staf keuangan. Wawancara meliputi proses yang terjadi saat ini dan data yang diperlukan untuk pembuatan sistem.

- c. Studi Pustaka

Studi Pustaka dilakukan dengan membaca dan mempelajari berbagai literatur yang berkaitan dengan masalah yang dibahas agar mendapatkan informasi yang mendalam. Sumber studi literatur mengenai analisis dan desain sebuah sistem serta mempelajari sistem informasi *invoice* yang sudah ada.

Langkah-langkah Pengembangan Sistem

Dalam melakukan penelitian, metode yang digunakan untuk membuat aplikasi menggunakan metode *waterfall*. Metode *waterfall* dilakukan dengan tahapan sebagai berikut :



Langkah-langkah Pengembangan Sistem

- a. Analisis Sistem

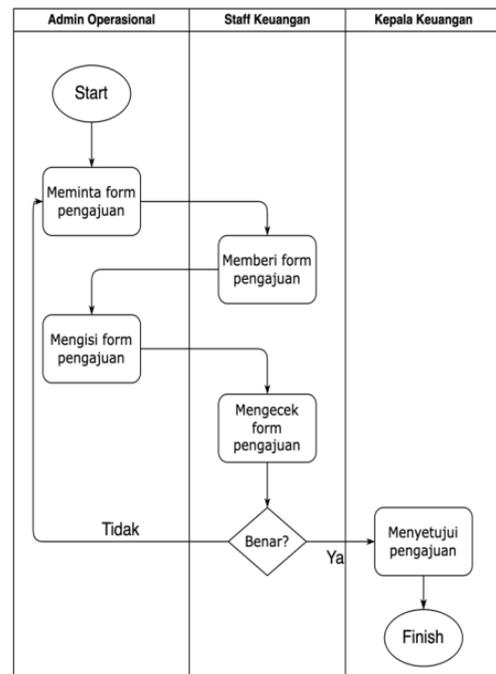
Pada tahap ini dilakukan pengumpulan data dan mempelajari literatur untuk mendukung pembuatan sistem. Tahap ini juga dilakukan pendefinisian dari data yang didapat kemudian dianalisis

- sesuai dengan kebutuhan yang harus dipenuhi untuk membangun sistem.
- b. Design Sistem
Tahap ini merupakan tahap pembuatan desain dari sistem dengan mendesain antarmuka, desain data, dan desain program. Desain sistem ini diharapkan dapat memberi gambaran kepada pembaca mengenai sistem yang dibangun.
 - c. Implementasi
Pembangunan sistem dibuat pada tahap ini. Sistem akan dibangun untuk membantu staf keuangan dalam pencatatan keuangan dan admin operasional untuk pengajuan pengadaan alat dan/atau bahan. Sistem ini akan dibangun sesuai dengan analisis dan desain yang sudah dilaksanakan sebelumnya.
 - d. Testing
Testing diperlukan untuk mengecek fungsionalitas sistem yang telah dibangun.

- f. Kepala keuangan menyetujui pengajuan anggaran.

Flow Map

Flow map pengajuan anggaran yang sedang berjalan di PT. Tirmuraya Jaya Lestari sebagai berikut:



Flow map sistem yang sedang berjalan

Analisa Sistem Yang Sedang Digunakan

Permasalahan yang ada pada PT.XYZ yaitu bagaimana pencatatan dan pengajuan anggaran dengan terstruktur dan rapi sehingga pencatatan keuangan lebih jelas dan pengajuan anggaran lebih mudah untuk dilaksanakan.

Prosedur pengajuan anggaran yang sedang berjalan di PT.XYZ adalah :

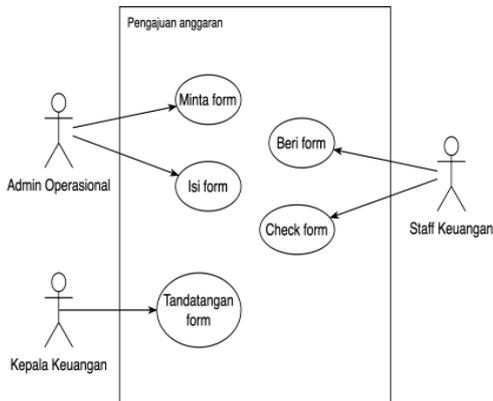
- a. Admin operasional meminta formula untuk pengajuan anggaran kepada staf keuangan.
- b. Admin operasional mengisi formula pengajuan anggaran.
- c. Admin operasional memberikan formula yang telah diisi.
- d. Staf keuangan mengecek formula yang sudah diisi oleh admin operasional.
- e. Staf keuangan memberikan formula tersebut kepada kepala keuangan.

Berdasarkan flowmap di atas, maka dapat diambil penjelasan pada sistem yang berjalan saat ini dimulai dari admin operasional meminta *form* untuk pengajuan *invoice* kepada staf keuangan, kemudian staf keuangan memberi *form* tersebut. Setelah itu, admin operasional mengisi *form* pengajuan dan staf keuangan mengecek *form* pengajuan yang sudah diisi, jika staf keuangan membenarkan pengajuan maka staf keuangan akan meminta persetujuan dari kepala keuangan, jika staf keuangan tidak membenarkan pengajuan maka *form* akan dikembalikan kepada admin operasional untuk mengisi *form* baru.

Use Case Diagram

Berikut ini adalah *use case diagram* yang menggambarkan pengajuan

anggaran pada PT.XYZ yang sedang berjalan.

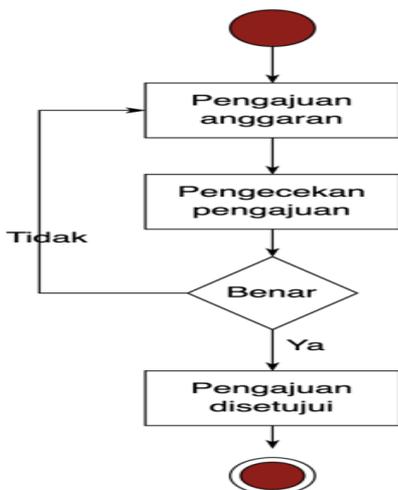


Use case diagram pada sistem yang sedang berjalan

Berdasarkan *use case diagram* diatas maka dapat disimpulkan bahwa untuk pengajuan anggaran, aksi yang dapat dilakukan oleh admin operasional adalah meminta form dan mengisi form pengajuan. Staff keuangan dapat memberi *form* dan mengecek *form* yang sudah diisi oleh admin operasional dan kepala keuangan dapat melakukan penandatanganan *form*.

Activity Diagram

Activity diagram di bawah ini menggambarkan aktivitas yang terjadi pada saat pengajuan anggaran pada PT.XYZ

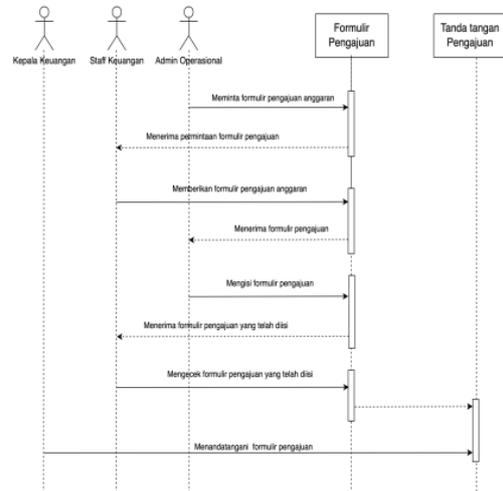


Activity diagram pada sistem yang sedang berjalan

Penjelasan *activity diagram* pada sistem yang sedang berjalan di PT.XYZ dengan urutan pengajuan anggaran, kemudian pengecekan pengajuan. Jika pengajuan dinilai benar maka pengajuan akan disetujui dan jika pengajuan dinilai tidak benar maka harus dilakukan pengajuan ulang.

Sequence Diagram

Berikut ini adalah *sequence diagram* yang menggambarkan pengajuan anggaran pada PT.XYZ yang sedang berjalan.



Sequence diagram pada sistem yang sedang berjalan

Berdasarkan *sequence diagram* yang sudah digambarkan tersebut maka dapat dilihat hal pertama yang dilakukan adalah admin operasional meminta formulir untuk pengajuan *invoice* kepada staf keuangan dan staf keuangan menerima permintaan formulir pengajuan. Kemudian staf keuangan memberi formulir tersebut dan admin operasional akan mendapatkan formulir yang sudah diberikan. Setelah itu, admin operasional mengisi formulir pengajuan dan staf keuangan akan menerima formulir pengajuan yang telah diisi. Staf keuangan akan meminta persetujuan dari kepala keuangan. Langkah terakhir

adalah kepala keuangan akan menandatangani formulir pengajuan.

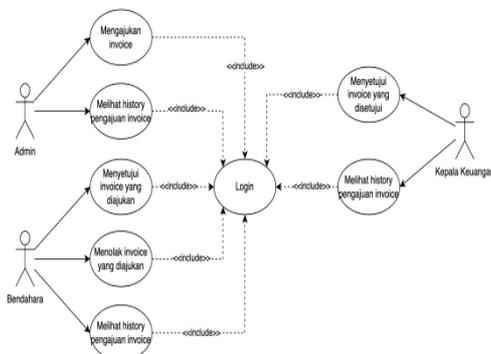
Implementasi Sistem

Implementasi sistem bertujuan untuk menunjang hasil rancangan dan analisis pada sistem. Implementasi dilakukan berdasarkan perancangan dan analisis yang sudah dibuat. Sistem akan dibuat berdasarkan hasil rancangan dan analisis kebutuhan.

Berdasarkan analisis kebutuhan maka dirancang sistem pengajuan *invoice online* berbasis *website* untuk PT.XYZ agar pengajuan *invoice* lebih efisien dan terarah.

Use Case Diagram Yang Diusulkan

Berdasarkan hasil Analisa yang telah dijalani maka *use case diagram* yang diusulkan untuk PT.XYZ adalah sebagai berikut.



Use case diagram yang diusulkan

Berdasarkan *use case diagram* di atas maka dapat disimpulkan bahwa untuk pengajuan anggaran, aksi yang dapat dilakukan oleh admin operasional adalah mengajukan *invoice* dan melihat *history* pengajuan *invoice*. Bendahara dapat menyetujui *invoice* yang diajukan, menolak *invoice* yang diajukan, dan melihat *history* pengajuan *invoice*. Kepala keuangan dapat menyetujui *invoice* yang sebelumnya telah disetujui oleh bendahara dan melihat *history* pengajuan *invoice*.

Use Case Scenario

Use case scenario digunakan untuk memudahkan analisa *scenario* yang akan digunakan pada fase selanjutnya dengan melakukan penilaian terhadap *scenario* tersebut.

Use case scenario pengajuan *invoice* pada role admin

Nama Use Case	Pengajuan Invoice	
Aktor	Admin	
Aksi Aktor	Reaksi Sistem	
Skenario Normal		
1. Mengakses halaman beranda		
	2. Menampilkan <i>invoice</i> semua yang pernah diajukan	
3. Klik tombol Tambah Pengajuan		
	4. Menampilkan untuk <i>form</i> pengajuan <i>invoice</i>	
5. Mengisi data pada <i>form</i> sesuai dengan kebutuhan dan klik <i>submit</i>		
	6. Menampilkan <i>invoice</i> di <i>submit</i>	

Tabel di atas merupakan tabel *use case scenario* untuk pengajuan *invoice* oleh admin.

Use case scenario melihat history invoice pada role admin

Nama Use Case	Melihat History Pengajuan Invoice
Aktor	Admin
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	
	2. Menampilkan invoice semua yang pernah diajukan
3. Pilih nomor invoice yang ingin dilihat at history-nya	
	4. Menampilkan lari data history yang dipilih

Tabel di atas merupakan tabel *use case scenario* untuk melihat *history* dari pengajuan *invoice*.

Use case scenario menghapus pengajuan invoice pada role admin

Nama Use Case	Hapus Pengajuan Invoice
Aktor	Admin
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	

	2. Menampilkan semua invoice yang pernah diajukan
3. Klik tombol "Hapus" yang terdapat pada kolom <i>action</i> berdasarkan <i>row</i> data <i>invoice</i> yang ingin dihapus	
	4. Menampilkan <i>pop up</i> validasi untuk penghapusan data
5. Memilih "Ya" pada <i>pop up</i> yang muncul	
	6. Invoice berhasil dihapus

Tabel di atas merupakan tabel *use case scenario* untuk melihat menghapus pengajuan *invoice* oleh admin.

Use case scenario edit pengajuan invoice pada role admin

Nama Use Case	Edit Pengajuan Invoice
Aktor	Admin
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	
	2. Menampilkan invoice semua yang pernah diajukan
3. Klik tombol "Edit" yang terdapat pada kolom	

<i>action</i> berdasarkan <i>row</i> data <i>invoice</i> yang ingin dihapus	
	4. Menampilkan <i>form</i>
5. Mengedit isi yang <i>form</i> dan klik <i>save</i>	
	6. <i>Invoice</i> berhasil diubah

Tabel di atas merupakan tabel *use case scenario* untuk *edit* pengajuan *invoice* oleh admin.

Use case scenario menyetujui/menolak *invoice* pada *role* bendahara

Nama Use Case	Menyetujui/Menolak <i>Invoice</i> Yang Diajukan
Aktor	Bendahara
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	
	2. Menampilkan semua <i>invoice</i> yang telah diajukan oleh admin
3. Pilih nomor <i>invoice</i> yang ingin dilihat untuk menampilkan detail <i>invoice</i>	

	4. Menampilkan detail <i>invoice</i>
5. Menyetujui/Menolak <i>invoice</i> yang diajukan (tambahkan catatan jika perlu)	
	6. Jika disetujui maka status pengajuan <i>invoice</i> menjadi <i>approve</i> dan <i>invoice</i> akan ditampilkan pada kepala keuangan. Jika ditolak maka status akan berubah menjadi <i>reject</i>

Tabel di atas merupakan tabel *use case scenario* untuk menyetujui atau menolak pengajuan *invoice* oleh bendahara.

Use case scenario melihat *report* pengajuan pada *role* bendahara

Nama Use Case	Melihat <i>Report</i> Pengajuan
Aktor	Bendahara
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	

	2. Menampilkan semua <i>invoice</i> yang telah diajukan oleh admin
3. Pilih <i>Report</i> pengajuan	
	4. <i>report</i> Menampilkan detail pengajuan per bulan

Tabel di atas merupakan tabel *use case scenario* untuk melihat *report* dari pengajuan *invoice* oleh bendahara.

Use case scenario melihat history *invoice* pada *role* bendahara

Nama Use Case	Melihat <i>History</i> Pengajuan <i>Invoice</i>
Aktor	Bendahara
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	
	2. Menampilkan <i>invoice</i> semua yang pernah diajukan
3. Pilih nomor <i>invoice</i> yang ingin dilihat <i>hist</i>	

<i>ory</i> - <i>nya</i>	
	4. Menampilkan data <i>history</i> yang dipilih

Tabel di atas merupakan tabel *use case scenario* untuk melihat *history* dari pengajuan *invoice* oleh bendahara.

Use case scenario melihat *history* pengajuan *invoice* pada *role* kepala keuangan

Nama Use Case	Melihat <i>History</i> Pengajuan <i>Invoice</i>
Aktor	Kepala Keuangan
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	
	2. Menampilkan semua yang pernah diajukan <i>invoice</i>
3. Pilih nomor <i>invoice</i> yang ingin dilihat <i>history</i> - <i>nya</i>	

	4. Menampilkan <i>history</i> yang dipilih
--	--

Tabel di atas merupakan tabel *use case scenario* untuk melihat *history* dari pengajuan *invoice* oleh kepala keuangan.

Use case scenario menyetujui pengajuan *invoice* pada *role* kepala keuangan

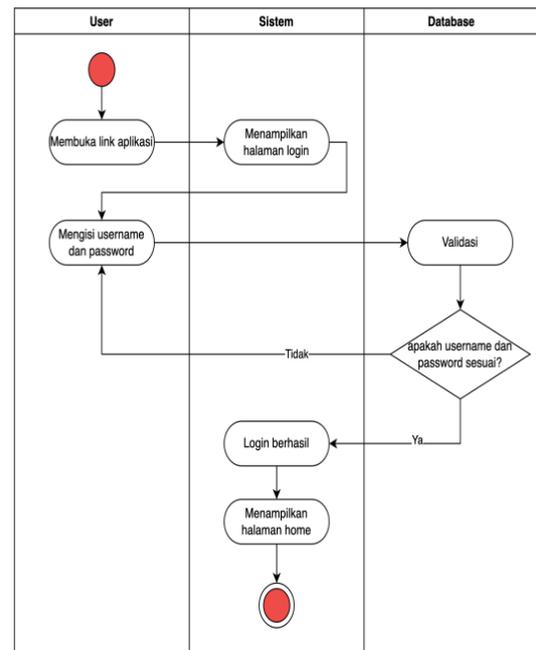
Nama Use Case	Menyetujui <i>Invoice</i> Yang Diajukan
Aktor	Kepala Keuangan
Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Mengakses halaman beranda	
	2. Menampilkan semua <i>invoice</i> yang telah disetujui oleh admin
3. Klik tombol yang ada di kolom <i>action</i> untuk menampilkan detail <i>invoice</i>	
	4. Menampilkan detail <i>invoice</i> yang dipilih
5. Menyetujui <i>invoice</i> yang diajukan	

	6. Mengubah status pengajuan <i>invoice</i> menjadi diketahui oleh kepala keuangan
--	--

Tabel di atas merupakan tabel *use case scenario* untuk menyetujui pengajuan *invoice* oleh kepala keuangan.

Activity Diagram Login

Activity diagram yang menerangkan alur proses untuk aktivitas *login user* dapat dilihat pada gambar di bawah.



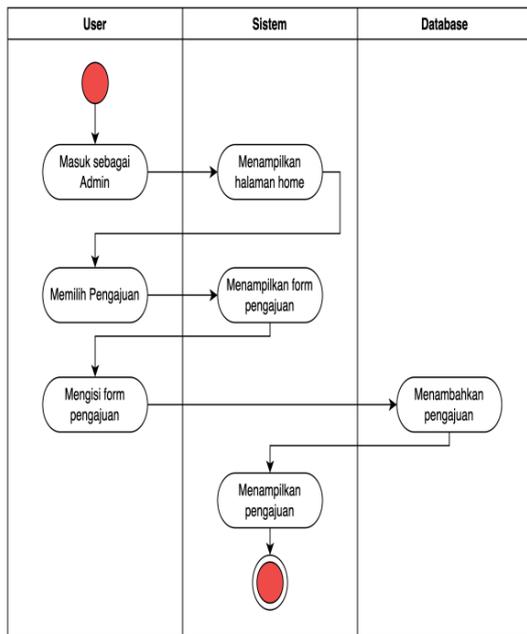
Activity diagram login

Diagram aktivitas di atas menerangkan alur proses untuk aktivitas *login*. Aktivitas dimulai *user* membuka *link* aplikasi. Kemudian sistem akan menampilkan halaman login, setelah itu *user* akan mengisi *username* dan *password* yang sesuai, kemudian *database* akan mengecek apakah data yang dimasukkan *user* sudah sesuai atau

belum. Jika *username* dan *password* cocok, maka sistem akan menampilkan halaman *home*.

Activity Diagram Mengajukan Invoice

Activity diagram yang menerangkan alur proses untuk aktivitas mengajukan *invoice* user dapat dilihat pada gambar di bawah.



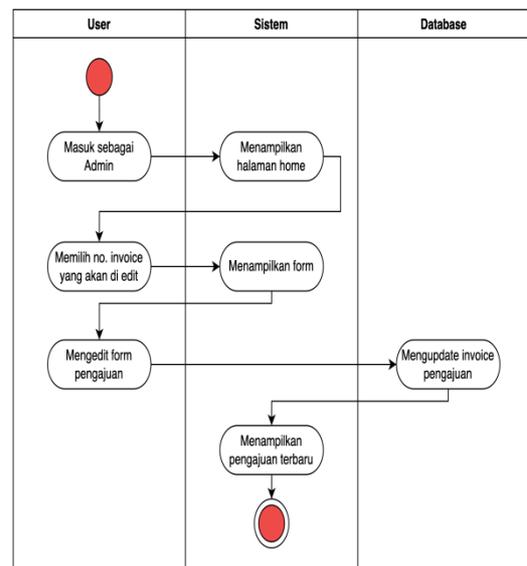
Activity Diagram Mengajukan Invoice

Diagram di atas menerangkan alur untuk pengajuan *invoice* oleh *user* admin. Langkah pertama yang harus dilakukan adalah *user* login sebagai admin. Kemudian sistem akan menampilkan halaman *home*. *User* memilih halaman pengajuan agar sistem dapat menampilkan *form* pengajuan *invoice*. Setelah itu *user* mengisi *form* pengajuan tersebut dan jika di *submit* maka data akan disimpan di *database* dan sistem akan menampilkan pengajuan yang telah di-*submit* tadi.

Activity Diagram Edit Invoice

Activity diagram yang menerangkan alur proses untuk aktivitas mengajukan mengubah isi dari *invoice* yang sudah diajukan sebelumnya. Pada

proses ini hanya dapat dilakukan jika *invoice* belum disetujui oleh Bendahara. Jika *invoice* sudah disetujui oleh Bendahara maka *invoice* tidak lagi dapat diubah.

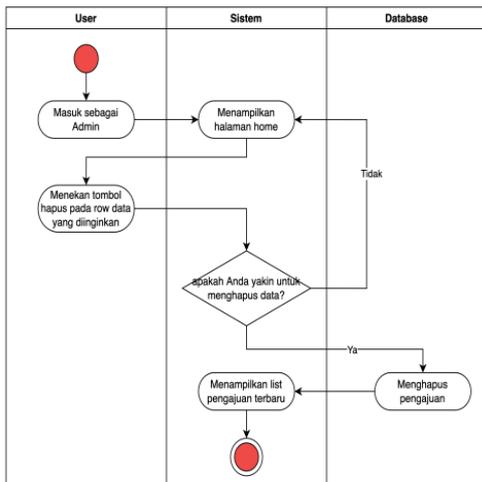


Activity diagram edit invoice

Proses ini dimulai dari *user login* sebagai Admin yang kemudian sistem akan menampilkan halaman *home* yang berisi *invoice* yang pernah diajukan. Kemudian *user* memilih *invoice* yang akan diubah, maka sistem akan menampilkan *form*. Setelah itu *user* mengubah isi *form* tersebut kemudian di simpan, maka *database* akan meng-*update* isi *invoice* dan sistem akan menampilkan pengajuan terbaru.

Activity Diagram Hapus Invoice

Activity diagram yang menggambarkan alur proses untuk aktivitas penghapusan *invoice* yang sudah diajukan sebelumnya. Pada proses ini hanya dapat dilakukan jika *invoice* belum disetujui oleh Bendahara. Jika *invoice* sudah disetujui oleh Bendahara maka *invoice* tidak lagi dapat dihapus.

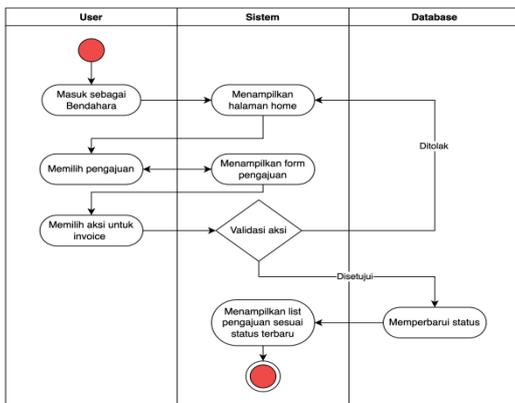


Activity diagram hapus invoice

Aktivitas ini dimulai dari *user* masuk sebagai Admin maka sistem akan menampilkan halaman home. Setelah itu user menekan tombol “hapus” yang ada di kolom *action* pada *row* data yang diinginkan. Kemudian sistem akan menampilkan *pop up* untuk validasi penghapusan. Jika *user* memilih Ya maka *invoice* akan dihapus dan sistem akan menampilkan *list* pengajuan terbaru. Jika memilih Tidak, maka sistem akan mengembalikan ke halaman *home*.

Activity Diagram Menyetujui atau Menolak Pengajuan Invoice

Activity diagram di bawah ini menggambarkan alur proses untuk menyetujui/menolak pengajuan *invoice*.

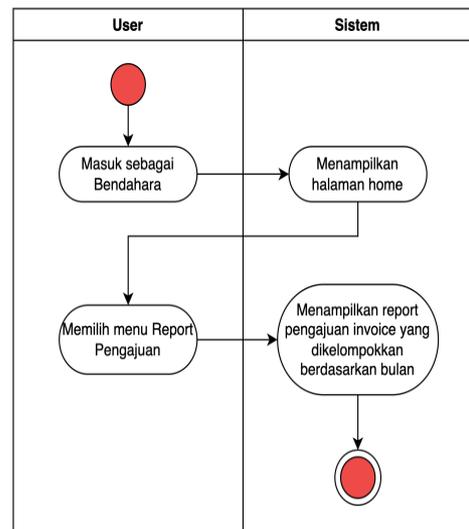


Activity diagram menyetujui/menolak invoice

Aktivitas ini dimulai dari *user* masuk sebagai Bendahara, maka sistem akan menampilkan halaman *home*. Setelah itu *user* memilih pengajuan. Kemudian sistem akan menampilkan *form* pengajuan. *User* memilih aksi untuk *invoice* tersebut dan sistem akan memvalidasi aksi yang dilakukan oleh *user*. Jika *user* menyetujui pengajuan maka *database* akan memperbarui status dan sistem akan menampilkan *list* pengajuan sesuai status terbaru. Jika user menolak pengajuan maka sistem akan mengembalikan ke halaman *home*.

Activity Diagram Melihat Report Pengajuan Invoice

Activity diagram yang menggambarkan alur proses untuk aktivitas melihat *report* pengajuan *invoice*, dapat dilihat pada gambar di bawah.

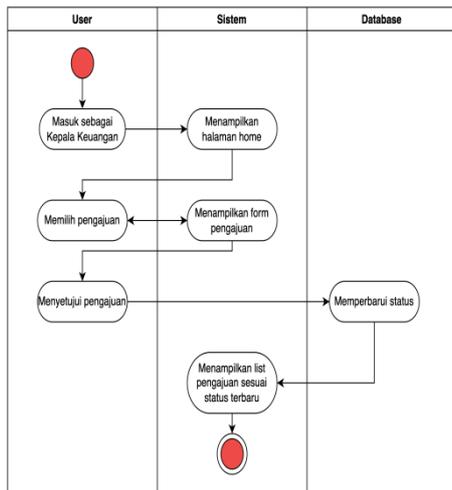


Activity diagram melihat report pengajuan invoice

Proses ini dimulai dari *user* login sebagai Bendahara dan sistem akan menampilkan halaman *home*. Kemudian *user* memilih menu *report* pengajuan dan sistem akan menampilkan *report* pengajuan *invoice* yang dikelompokkan berdasarkan bulan.

Activity Diagram Menyetujui Pengajuan Invoice

Activity diagram yang menggambarkan alur proses untuk aktivitas menyetujui pengajuan invoice, dapat dilihat pada gambar di bawah.

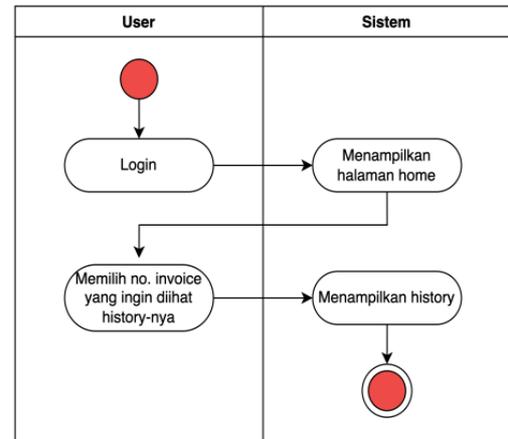


Activity diagram meyetujui pengajuan invoice

Proses ini dimulai dari user login sebagai Kepala Keuangan dan sistem akan menampilkan halaman home. Kemudian user memilih pengajuan dan sistem akan menampilkan form pengajuan invoice yang telah disetujui oleh Bendahara. Setelah itu user menyetujui pengajuan maka database akan memperbarui status dan sistem akan menampilkan list pengajuan sesuai status terbaru.

Activity Diagram Melihat History Invoice

Activity diagram di bawah ini menggambarkan alur proses untuk melihat history pengajuan invoice.



Activity diagram melihat history invoice

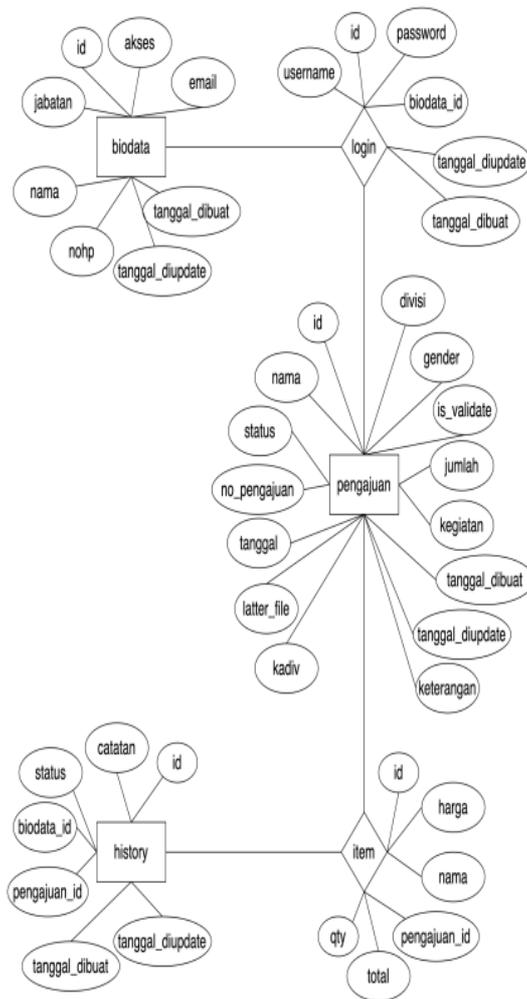
Aktivitas ini dimulai dari user masuk, maka sistem akan menampilkan halaman home. Setelah itu user menekan No. Invoice pada row data yang ingin ditampilkan history-nya. Kemudian sistem akan menampilkan halaman yang berisikan history dari invoice yang dipilih.

Rancangan Database Yang Diusulkan

Rancangan skema database pada sistem ini berdasarkan desain use case dan activity diagram yang sudah diusulkan pada poin sebelumnya. Database dibuat berdasarkan fitur seta interaksi antar aktor dan juga antar use case. Database akan digambarkan dalam bentuk Entity Relationship Diagram (ERD) dan juga tabel.

Entity Relationship Diagram (ERD)

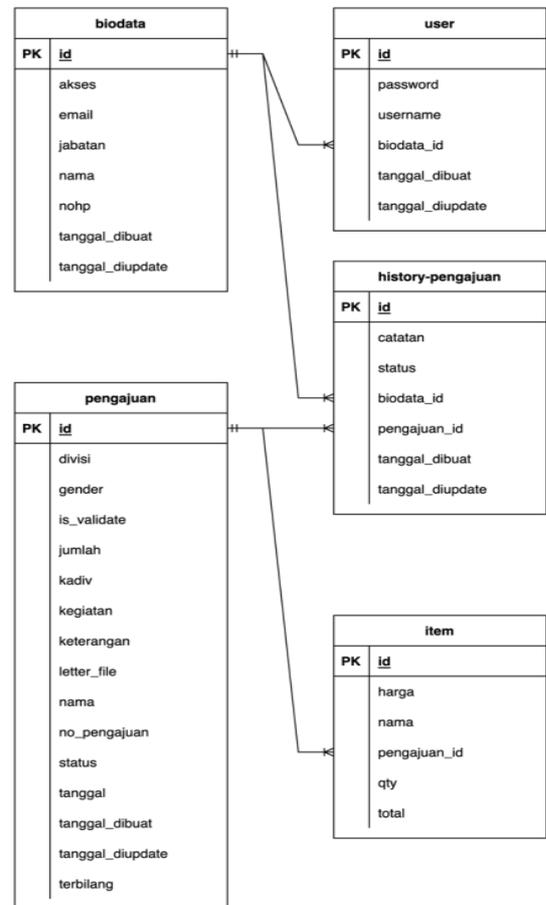
Entity Relationship Diagram atau biasa disebut ERD ini berfungsi untuk menggambarkan hubungan dari setiap tabel yang ada pada database. ERD memudahkan user untuk mengatur tujuan dan arah yang ada pada setiap database. Pada hal ini, penulis akan menggambarkan ERD yang diusulkan untuk sistem ini seperti gambar di bawah ini.



Entity Relationship Diagram (ERD)

Class Diagram

Class diagram menggambarkan struktur sistem dari sisi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas dan mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut. Metode atau operasi adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Class diagram dapat digambarkan sebagai berikut.



Class Diagram

Keterangan :

- One to One : Menunjukkan hubungan satu ke satu
- One to Many : Menunjukkan hubungan satu ke banyak
- Many to One : Menunjukkan hubungan banyak ke satu
- Many to Many : Menunjukkan hubungan banyak ke banyak

Kamus Data (Data Dictionary)

Kamus data diperlukan untuk menjabarkan setiap tabel yang ada pada database. Kamus data ini juga sebagai acuan dalam pembuatan database untuk sistem ini dan akan dituangkan dalam bentuk tabel seperti berikut.

- Nama database : timuraya
- Tabel biodata
Tabel biodata merupakan entitas untuk menyimpan data yang

berisikan biodata *user*. Tabel ini terhubung dengan tabel relasi dengan menggunakan *field* id. Pada tabel terdapat delapan *field*, yaitu id sebagai *primary key*, akses, email, jabatan, nama, nohp, tanggal_dibuat, dan tanggal_diupdate. Tabel biodata akan direpresentasikan pada tabel 4.10 di bawah.

Tabel biodata pada database

No.	Nama <i>Fields</i>	Tipe	Keterangan
1.	id	Bigint	Primary Key
2.	akses	Varchar	
3.	email	Varchar	
4.	jabatan	Varchar	
5.	nama	Varchar	
6.	nohp	Varchar	
7.	tanggal_dibuat	Datetime	
8.	tanggal_diupdate	Datetime	

- b. Tabel history-pengajuan
Tabel history-pengajuan merupakan entitas untuk menyimpan data yang berisikan *history* pengajuan *invoice*. Pada tabel terdapat tujuh *field*, yaitu id sebagai *primary key*, catatan, status, biodata_id, pengajuan_id, tanggal_dibuat, dan tanggal_diupdate. Tabel history pengajuan akan direpresentasikan pada tabel 4.11 di bawah.

Tabel history-pengajuan pada database

No.	Nama <i>Fields</i>	Tipe	Keterangan
1.	id	Bigint	<i>Primary Key</i>
2.	catatan	Varchar	
3.	status	Varchar	
4.	biodata_id	Bigint	
5.	pengajuan_id	Bigint	

6.	tanggal_dibuat	Datetime	
7.	tanggal_diupdate	Datetime	

- c. Tabel item
Tabel item merupakan entitas untuk menyimpan data yang berisikan item yang ada pada *invoice*. Pada tabel terdapat enam *field*, yaitu id sebagai *primary key*, harga, nama, qty, pengajuan_id, dan total.

Tabel item akan direpresentasikan pada tabel 4.12 di bawah.

Tabel item pada database

No.	Nama <i>Fields</i>	Tipe	Keterangan
1.	id	Bigint	<i>Primary Key</i>
2.	harga	Numer ic	
3.	nama	Varch ar	
4.	qty	Int	
5.	pengajuan_id	Bigint	
6.	total	Numer ic	

- d. Tabel pengajuan
Tabel pengajuan merupakan entitas untuk menyimpan data yang berisikan pengajuan *invoice*. Pada tabel ini terdapat 16 *field*, yaitu id sebagai *primary key*, divisi, gender, is_validate, jumlah, kadv, kegiatan keterangan, letter_file, nama, no_pengajuan, status, tanggal, terbilang, tanggal_dibuat, dan tanggal_diupdate. Tabel pengajuan akan direpresentasikan pada tabel 4.13 di bawah.

Tabel pengajuan pada database

No.	Nama <i>Fields</i>	Tipe	Keterangan
1.	id	Bigint	<i>Primary Key</i>
2.	divisi	Varchar	
3.	gender	Varchar	
4.	Is_validate	Boolean	
5.	jumlah	numeric	
6.	kadiv	Varchar	
7.	kegiatan	Varchar	
8.	Keterangan	Varchar	
9.	Latter_file	Varchar	
10.	Nama	Varchar	
11.	No_pengajuan	Varchar	
12.	Status	Varchar	
13.	Tanggal	Datetime	
14.	Tanggal_dibuat	Datetime	
15.	Tanggal_diuupdate	Datetime	
16.	terbilang	Varchar	

e. Tabel user

Tabel user merupakan entitas untuk menyimpan data yang berisikan informasi *user*. Pada tabel terdapat enam *field*, yaitu id sebagai *primary key*, password, username, biodata_id, tanggal_dibuat, dan tanggal_diuupdate. Tabel user akan direpresentasikan pada tabel 4.14 di bawah.

Tabel user pada database

No.	Nama <i>Fields</i>	Tipe	Keterangan
1.	id	Bigint	<i>Primary Key</i>
2.	password	Varchar	
3.	username	Varchar	
4.	biodata_id	Bigint	
5.	tanggal_dibuat	Datetime	
6.	tanggal_diuupdate	Datetime	

Tampilan Sistem Yang Dibuat

Tampilan sistem dibuat berdasarkan analisis yang sudah dilakukan. Proses pembuatan sistem ini dilakukan dengan mengimplementasikan analisis berdasarkan UML dan desain yang sudah dilakukan. Pada sistem ini setiap *user* memiliki hak akses yang berbeda-beda dan menyebabkan tampilan yang ditampilkan akan berbeda-beda juga pada setiap *user*. Tampilan sistem dibuat sebagai berikut.

Kesimpulan

Berdasarkan pembahasan mengenai sistem pengajuan *invoice online* pada PT.XYZ, maka dapat disimpulkan bahwa sistem mampu memvalidasi laporan terkait sistem informasi teknis operasional dengan mudah dan akurat. Selain itu, dengan adanya sistem ini, pendataan *invoice* sudah dapat dilakukan secara mekanis. Sistem ini juga lebih mudah mengetahui informasi perangkat yang dibutuhkan oleh pengguna dan pengguna juga dapat memiliki gambaran perangkat yang digunakan.

Sistem ini dirancang menggunakan Bahasa pemrograman java. Database yang digunakan pada perancangan sistem ini menggunakan PostgreSQL dan metode yang digunakan pada sistem ini adalah metode waterfall. Sistem telah

dirancang dan dibuat berdasarkan semua elemen yang digunakan sehingga sistem mampu untuk melakukan pengajuan *invoice* secara *online* berdasarkan kebutuhan perusahaan dan pengajuan dapat ditolak atau diterima berdasarkan hasil pengecekan yang dilakukan oleh Bendahara.

Sistem ini diharapkan dapat membantu dalam pengajuan *invoice* pada PT.XYZ agar pengajuan *invoice* lebih efisien. Dengan menggunakan sistem ini Admin, Bendahara, serta Kepala Keuangan akan mengurangi kesalahan pengajuan dan penyimpanan *form* pengajuan akan lebih aman.

Daftar Pustaka

- Bagus, P. 2018. "Analysis of A Priori Algorithm on Sales Transactions to Arrange Placement of Goods on Minimarket ." *International Journal of Engineering and Emerging Technology* 13-17.
- Pane, Syafril Fachri, and Yogi Aditya Saputra. 2020. *Big Data classification behavior menggunakan python*. Bandung: Kreatif Industri Nusantara.
- Ramadhan, Puji Sari, and Usti Fatimah S. Pane. 2018. *Mengenal Metode Sistem Pakar*. Sidoarjo: Uwais Inspirasi Indonesia.
- Setiadi, Ahmad, Y Yunita, and Ibung Prasetio Nugroho. 2019. "Sistem Pakar Diagnosa Penyakit Lambung Menggunakan Forward Chaining." *Pendidikan Informatika dan Sains* 19-31.
- Hidayat, Fendi. 2020. *Konsep Dasar Sistem Informasi Kesehatan*. Yoogyakarta: CV Budi Utama.
- Rumindo, and Muttaqin. 2019. *E-Commerce: Implementasi, Strategi, dan Inovasinya*. Medan: Yayasan Kita Menulis.
- Rizky, Muhammad. 2021. "Sistem Pendukung Keputusan Penentuan Badan Usaha Milik Desa Terbaik Menetapkan Metode VIKTOR." 2: 8.
- Andriani, A. 2021. *Pemrograman Sistem Pakar*. Yogyakarta: Mediakom.
- Haryantoko. 2018. *Sistem Informasi Keuangan Di Pertenakan Sapi "Madu Indah"*. Boyolali: STMIK AMIKOM YOGYAKARTA.
- Komariah. 2019. "Sistem Informasi Ternak Ayam Pada CV Hanan Jaya Berbasis WEB."
- Kurniawan, Arif. 2021. "Sistem Rekomendasi Produk Sepatu Dengan Menggunakan Metode Collaborative Filtering." *Seminar Nasional Teknologi Informasi dan Komunikasi*.
- Lesmana, Lido Sabda. 2020. "Sistem Pakar Dengan Metode Forward Chaining Untuk Diagnosa Pasien Yang Terinfeksi Virus MERS Cov (Studi Kasus RSUP M. Djamil Padang)." *CBIS Journal*.
- Rosid. 2021. "Implementasi JSON untuk Minimasi Penggunaan Jumlah Kolom suatu Tabel pada Database PostgreSQL." *Journal Of Informatics, Network, and Computer* 33-42.
- Yakub. 2018. *Pengantar Sistem Informasi*. Yogyakarta: Graha Ilmu.
- Sarwono, Jonathan, and Prihartono. 2018. *Perdagangan Online: Cara Bisnis di Internet*. Jakarta: PT Elex Media Komputindo.
- Mustakin. 2018. *Aplikasi E-Commerce Dalam Sistem Penjualan DiMboy's Cell Demak*. Semarang: STEKOM.