



## Penerapan *rapid application development* dalam rancang bangun sistem pemijaman aset PT Ukirama Solusi Indonesia

### *Implementation of rapid application development in the design and construction of PT Ukirama Solusi Indonesia's asset lending system*

Yohana Christela Oktaviani<sup>1</sup>, Yulia Wahyuningsih<sup>1\*</sup>, Fernandi Mahendrasusila<sup>2</sup>

<sup>1\*</sup> Ilmu Informatika, Fakultas Teknik, Universitas Katolik Darma Cendika, Jl. Dr. Ir. H. Soekarno No. 201, Klampis Ngasem, Sukolilo, Surabaya, Indonesia

<sup>2</sup> PT Elevenia Sinergi Prima Nusantara, Jl. Jenderal Sudirman Kav. 1, RT.1/RW.8, Karet Tengsin, Jakarta Pusat, Indonesia

#### INFORMASI ARTIKEL

#### ABSTRAK

##### Article History:

Submission: 24-11-2023

Revised: 07-12-2023

Accepted: 14-12-2023

##### Kata Kunci:

Teknologi; RAD; *blackbox testing*; sistem pemijaman aset.

##### Keywords:

Technology; RAD; *blackbox testing*; *asset lending system*.

##### \* Korespondensi:

Yulia Wahyuningsih  
yulia@ukdc.ac.id

Aset merupakan suatu fasilitas yang digunakan untuk meningkatkan produktivitas perusahaan. Penggunaan aset pada perusahaan perlu dilakukan pencatatan untuk menghindari kemungkinan terjadinya kerusakan maupun kehilangan aset pada perusahaan. Proses pencatatan ini tentunya akan sangat memberatkan jika dilakukan secara konvensional. Oleh sebab itu, perusahaan membutuhkan suatu sistem yang dapat mencatat pemijaman serta pengembalian aset. Untuk membangun sistem yang bersifat *customized* lazimnya membutuhkan waktu yang cukup lama, yakni sekitar 180 hari. Dalam hal ini, untuk membangun sistem pemijaman aset, diperlukan suatu model rancang bangun sistem yang lebih cepat dan *applicable*. Sistem pemijaman aset ini dirancang dan dibangun dengan menerapkan model *Rapid Application Development* (RAD). Penerapan model RAD pada sistem pemijaman aset ini diuji coba menggunakan metode *Blackbox Testing*. Penelitian yang telah dilakukan menunjukkan bahwa model RAD mampu mempercepat tahap pembangunan sistem dimana proses pembangunan berlangsung selama  $\pm 60$  hari kerja dengan pengerjaan 5 jam/hari. Selain mempercepat tahap pembangunan sistem, model RAD juga menghasilkan sistem dengan fungsionalitas yang baik dan sesuai dengan *requirement* yang telah disusun bersama *user*. Hal tersebut ditunjukkan dari hasil pengujian sistem dengan *feedback* 100% berhasil.

#### ABSTRACT

*Asset is a facility that is used to increase company productivity. Assets utilization needs to be recorded to avoid the possibility of damage or loss of assets in the company. This process would certainly be very burdensome if done conventionally. Therefore, companies need a system that can record loans and return assets. To build a customized system usually takes quite a long time, namely around 180 days. In this case, to build an asset lending system, a system design model that is faster and more applicable is needed. This asset lending system was designed and built using the Rapid Application Development (RAD) model. Implementation of the RAD model to this asset lending system was tested using the Blackbox Testing method. Research that has been carried out shows that the RAD model is able to speed up the system development stage where the development process takes  $\pm 60$  working days with 5 hours of work per day. Apart from speeding up the system development stage, the RAD model also produces a system with good functionality and in*



*accordance with the requirements that have been prepared with the user.  
This is shown by the results of system testing with 100% successful feedback.*

---

## 1. PENDAHULUAN

Kemajuan teknologi dewasa ini memiliki peranan penting dalam setiap segi kehidupan, baik dalam instansi pendidikan, pemerintahan, kesehatan, serta perusahaan. Dalam perusahaan sendiri, teknologi ini berperan dalam meningkatkan efisiensi, efektivitas, serta produktivitas perusahaan. *Encyclopedia Britania* mengungkapkan bahwa produktivitas merupakan rasio dari hasil yang dicapai dengan usaha yang dikeluarkan untuk mencapai suatu tujuan [1]. Bentuk usaha eksternal yang dikeluarkan oleh perusahaan dapat berupa barang dan jasa, sedangkan bentuk usaha internal yang dikeluarkan oleh perusahaan salah satunya ialah pengelolaan aset perusahaan. Aset perusahaan ini dapat berupa sumber daya manusia, sumber daya alam, jaringan relasi, serta alat pendukung operasional perusahaan [2]. Dalam mengelola alat pendukung operasional perusahaan, perlu dilakukan pencatatan peminjaman aset. Hal tersebut dilakukan karena aset tetap bersifat relatif yang sewaktu-waktu dapat mengalami perubahan, baik pengurangan maupun penambahan. Selain itu, pencatatan peminjaman aset perlu dilakukan untuk menghindari kemungkinan terjadinya kerusakan dan kehilangan barang oleh oknum yang tidak bertanggung jawab.

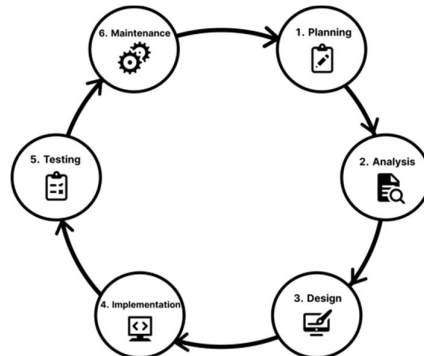
Di era modern seperti sekarang ini, perusahaan dan instansi lainnya, seperti pendidikan dan pemerintahan menerapkan berbagai cara untuk melakukan pencatatan peminjaman aset, mulai dari cara konvensional, *semi-computerized*, hingga menggunakan sistem yang dirancang khusus untuk melakukan pencatatan peminjaman aset. Penerapan cara konvensional dan *semi-computerized* pada pencatatan peminjaman aset masih dimungkinkan karena jika membeli sistem yang sudah jadi, peluang untuk sistem tersebut dapat dibuat secara *custom* sangatlah kecil, sehingga belum tentu dapat memenuhi kebutuhan instansi. Selain itu, waktu yang diperlukan untuk membangun sebuah sistem cukup lama, yakni minimal 180 hari [3]. Jangka waktu pembangunan sistem yang cukup lama ini dapat mengakibatkan terjadinya pembengkakan biaya. Oleh sebab itu, beberapa instansi memilih untuk menggunakan cara konvensional dan *semi-computerized*. Namun, beberapa peneliti terdahulu telah melakukan peralihan pada proses pencatatan peminjaman aset yang diterapkan dalam berbagai bidang. Anggita Puspaningrum, dalam penelitiannya mengembangkan media pencatatan peminjaman aset dari yang semula dilakukan secara konvensional menggunakan papan tulis menjadi sistem manajemen peminjaman dan pengelolaan aset berbasis website yang diterapkan pada Taman Budaya Jawa Tengah [4]. Metode konvensional lain yang terdapat pada aktivitas pencatatan peminjaman aset ialah penelitian pada Universitas Budi Luhur yang semula dilakukan dengan cara mengirimkan surat secara langsung menjadi sistem informasi peminjaman aset berbasis website [5]. Selain itu, terdapat pula penelitian terdahulu yang mengembangkan media pencatatan peminjaman aset yang semula dalam bentuk *semi-computerized* menggunakan *Microsoft Excel* menjadi sistem informasi berbasis *website*, seperti yang diterapkan pada Kantor Lapas II A Cibinong yang membangun sistem informasi peminjaman aset Kantor Lapas [6], PT Bangunindo Teknusa Jaya yang membangun sistem informasi dan diintegrasikan dengan *database* [7], dan PT Metis Teknologi Corporindo yang membangun sistem informasi manajemen aset mulai dari proses pengadaan aset baru, pembelian, hingga perhitungan penyusutan nilai aset [8]. Selain diterapkan dalam bentuk *semi-computerized*, pencatatan peminjaman aset juga dapat dilakukan dengan menerapkan teknologi *Radio Frequency Identification* (RFID), seperti penelitian yang dilakukan pada sistem peminjaman barang perusahaan [2], dan sistem peminjaman aset laboratorium [9].

Dalam hal ini, PT Ukirama Solusi Indonesia yang merupakan perusahaan yang bergerak di bidang teknologi membutuhkan suatu sistem yang dirancang khusus untuk melakukan pencatatan peminjaman aset perusahaan. Ukirama membutuhkan sistem pencatatan peminjaman aset yang dibangun secara efisien agar dapat digunakan oleh perusahaan sesegera mungkin. Sistem peminjaman aset ini akan dibangun dengan bahasa pemrograman Ruby dengan *framework* Rails. Terlepas dari metode pencatatan konvensional, *semi-computerized* maupun sistem yang dibangun dalam waktu yang lama, sistem ini dibangun menggunakan model *Rapid Application Development* (RAD) dan diuji dengan menggunakan metode *Blackbox Testing*. Model ini dapat digunakan untuk membangun sistem yang dirancang secara *custom* dan efisien karena

selama proses desain dan implementasi berlangsung, terdapat interaksi yang intens antara *user* dengan *developer* sehingga *user* dapat memberikan *feedback* secara terus menerus sampai ditemukan hasil yang sesuai dengan kebutuhan *user*. Hal tersebut membuat proses pembangunan sistem berlangsung dalam waktu yang lebih singkat dibandingkan dengan model lainnya, yakni dalam rentang waktu 30-90 hari [10].

## 2. METODE

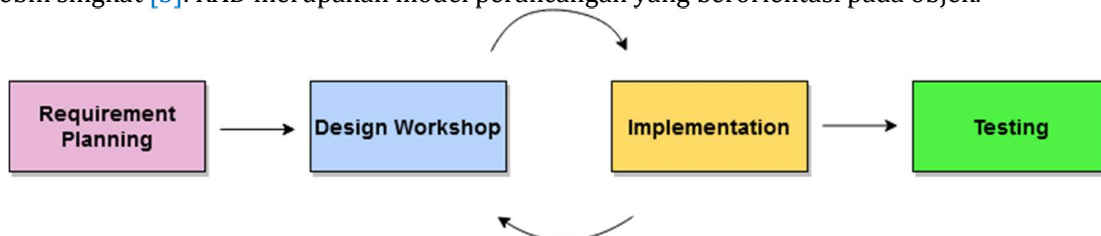
Proses pembuatan sistem peminjaman aset ini dirancang dan dibangun menggunakan metode *System Development Life Cycle* (SDLC), yaitu suatu proses perancangan sistem yang diterapkan ketika sistem pertama kali dibuat. Secara umum, SDLC ini terdiri dari tiga kategori, yakni *traditional system life cycle*, *prototyping*, serta *object-oriented system life cycle* [3].



Gambar 1. Alur Metode SDLC

Metode SDLC ini bersifat inkremental dan terstruktur [11] dalam penerapannya serta terdiri dari enam tahapan seperti ditunjukkan pada Gambar 1, antara lain *planning*, *analysis*, *design*, *implementation*, *testing*, dan *maintenance* [10]. Metode SDLC memiliki berbagai jenis model dengan kategori yang berbeda. Salah satu model SDLC yang termasuk ke dalam kategori *Object Oriented System Life Cycle* adalah *Rapid Application Development* (RAD).

*Rapid Application Development* (RAD) merupakan model perancangan perangkat lunak yang dilakukan dalam waktu yang relatif singkat dan bersifat inkremental [12]. RAD merupakan hasil pengembangan dari model *waterfall* dengan versi perancangan yang lebih cepat karena RAD mengadopsi beberapa tahapan pada model *waterfall* tetapi tidak menerapkan semuanya. Beberapa tahapan yang terdapat pada model *waterfall* digabungkan dan dikompres menjadi satu rangkaian tahapan yang lebih *compact*. RAD bersifat inkremental dan melibatkan pengguna pada setiap prosesnya dimana perancangan dan pengembangan sistem dilakukan secara bertahap, hal ini memungkinkan tim *developer* untuk memperoleh *feedback* lebih awal dari *user*. Berbeda dengan model *waterfall* dan model *traditional system life cycle* lainnya yang menyelesaikan setiap tahap perancangan secara linear dan terurut. Proses perancangan sistem yang dilakukan secara inkremental dalam model RAD memudahkan *developer* untuk mengidentifikasi masalah atau *bug* lebih awal dan sistem menjadi lebih fleksibel terhadap perubahan [13]. Model RAD sendiri cocok untuk diterapkan pada sistem yang dirancang berdasarkan kebutuhan pengguna, berskala besar, dan membutuhkan waktu eksekusi yang lebih singkat [3]. RAD merupakan model perancangan yang berorientasi pada objek.



Gambar 2. Alur Model RAD

Gambar 2 menggambarkan model RAD yang terdiri dari empat tahapan, yakni *Requirement Planning*, *Design Workshop*, *Implementation*, dan *Testing*. Pada tahap *Design*

*Workshop* dan *Implementation* dapat terjadi beberapa kali perulangan yang disesuaikan dengan *feedback user* berdasarkan kebutuhannya. Pada saat proses perancangan, tahap RAD ini tidak dapat terlepas dari evaluasi *user*. Inilah salah satu aspek yang menyebabkan model RAD lebih cepat daripada model lainnya dimana *user* berperan secara intensif untuk memberikan *feedback* terhadap perancangan suatu sistem agar hasil yang diperoleh nantinya sesuai dengan kebutuhan *user*. Tahap *design workshop* pada model RAD tidak hanya mengacu pada desain UI/UX saja, melainkan juga desain fungsional sistem yang dirancang dengan menggunakan *Unified Modelling Language* (UML) dan desain rancangan *database* dengan *Conceptual Data Model* (CDM).

*Unified Modelling Language* (UML) merupakan *tools* yang berfungsi untuk menggambarkan pemodelan suatu sistem berorientasi objek [14]. Proses pemodelan sistem dengan menggunakan UML ini dapat membantu *user* dalam memberikan gambaran, rancangan, dan dokumentasi setiap kerangka pembangunan sistem secara lebih efektif dan terstruktur [15]. Pembuatan UML dapat mempermudah proses analisis dan desain perancangan system [16]. Sedangkan *Conceptual Data Model* atau CDM ini merupakan sebuah model perancangan *database* yang dibangun berdasarkan informasi yang diperoleh dari *user requirement* dalam suatu instansi [17]. Model ini digunakan untuk merepresentasikan kebutuhan data dalam *database*. CDM merupakan landasan dasar dalam membangun sebuah *Physical Data Model* yang lebih mendetail [18].

Dalam SDLC, pengujian merupakan tahap penting yang harus dilakukan, salah satunya dengan menggunakan metode pengujian *Blackbox Testing*. *Blackbox* merupakan metode pengujian yang bukan mengacu pada pengujian desain tampilan maupun kode program, melainkan pengujian fungsionalitas sistem untuk memastikan sistem yang dibangun telah sesuai dengan kebutuhan *user* dan dapat berjalan dengan baik [19]. Estimasi jumlah data yang diuji dapat ditentukan berdasarkan jumlah *field* data dan aturan pengisian data yang harus dipenuhi berdasarkan *requirement*. Tujuan dari pengujian ini adalah untuk mengetahui kekurangan sistem setelah sistem diimplementasikan untuk menghindari kesalahan pada sistem sebelum digunakan oleh *user* [20].

Berikut merupakan tahapan dalam implementasi model RAD pada sistem peminjaman aset perusahaan.

### 2.1 Requirement Planning

Merupakan tahap awal yang dilakukan dalam mengumpulkan data yang diperlukan untuk membangun sistem peminjaman aset. Pengumpulan data dilakukan melalui proses wawancara antara *Human Resource Development* (HRD) sebagai *user* dengan *Software Engineer* perusahaan. Berikut merupakan hasil *requirement* yang telah diperoleh melalui proses wawancara bersama *user*:

- a. Sistem peminjaman aset terdiri dari dua *role* pengguna, yakni admin dan *user*.
- b. Jenis aset yang dapat dipinjam dapat berupa jenis aset jangka pendek (TV, proyektor, printer) maupun aset jangka panjang (PC, laptop).
- c. Proses peminjaman aset harus melalui admin (admin memberikan persetujuan terlebih dahulu).
- d. Pengembalian barang dilakukan pada sistem yang sama dengan proses yang sama, yakni mengajukan pengembalian dan disetujui oleh admin.
- e. Admin berwenang untuk menyetujui atau tidak menyetujui pengajuan peminjaman aset (berdasarkan kebutuhan dan tingkat kepentingan).
- f. Admin dapat melihat semua data aset (master aset).
- g. Admin dapat mengelola (menambah, menyunting, dan menghapus) data aset.
- h. Admin dapat melihat semua data *user*.
- i. Admin dapat mengelola (menambah, menyunting, dan menghapus) data *user*.
- j. Admin dapat melihat dan mengelola semua data peminjaman dan pengembalian aset.
- k. *User* hanya dapat melihat data barang secara umum, seperti nama dan merk, tidak dengan kelengkapan data barangnya, seperti nomor seri barang.
- l. *User* dapat melakukan peminjaman secara mandiri melalui sistem.

- m. *User* hanya dapat melihat data barang yang *available* saja (barang tersedia dan belum dipinjam).
- n. *User* hanya dapat melihat data peminjaman dan pengembalian barang yang dimiliki oleh masing-masing *user*.

Berdasarkan data *requirement analysis* yang telah dilakukan, diperoleh rumusan perancangan halaman sistem sebagai berikut.

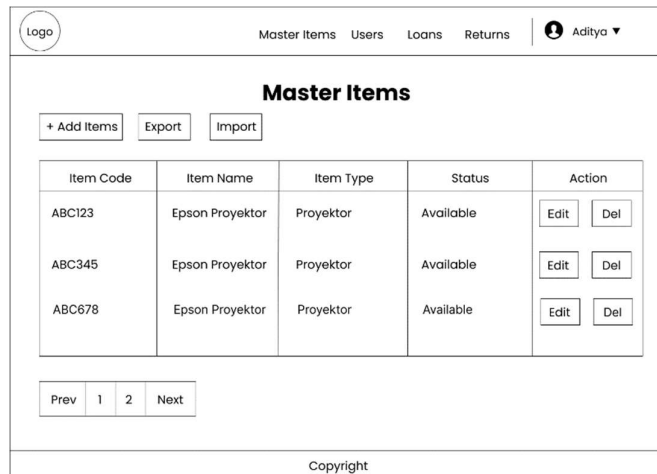
- a. *Landing page*
- b. *Index user* (master data *user*)
- c. *Login page* (admin & *user*)
- d. Form tambah dan ubah data aset dan *user*
- e. *Forgot password page*
- f. *Approval page* peminjaman aset
- g. *Sign up page*
- h. *Form* peminjaman dan pengembalian aset
- i. *Profile page*
- j. *Index* peminjaman dan pengembalian aset
- k. *Index* aset (master data aset)

## 2.2 *Design workshop*

Setelah dilakukan *user requirement analysis* dan merumuskan halaman yang diperlukan pada tahap *requirement planning*, langkah berikutnya yang harus dilakukan adalah merancang *design system*. Perancangan *design system* ini terbagi menjadi tiga tahapan, yakni perancangan antarmuka, perancangan diagram UML serta perancangan *database* yang akan digunakan.

### a. Perancangan antarmuka

Antarmuka pada rancang bangun sistem peminjaman aset ini dilakukan dengan menggunakan metode *prototyping*. Metode ini merupakan proses awal dalam merancang dan mengembangkan suatu sistem yang berfokus pada apa yang dilihat oleh *user* [21]. Metode ini bertujuan untuk menyatukan pemahaman antara *user* dengan *developer* selama proses perancangan berlangsung [22]. Dalam hal ini, sistem didesain dalam bentuk *low fidelity* berdasarkan *requirement* yang telah diperoleh dari *user* yang ditunjukkan pada Gambar 3.



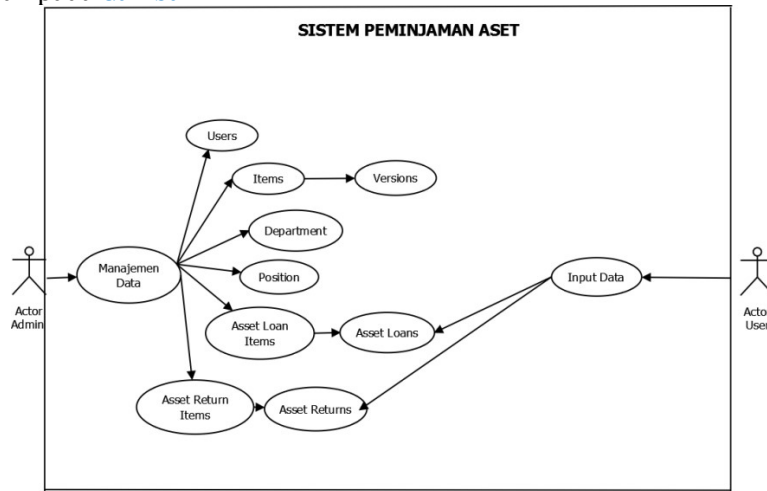
Gambar 3. *Low fidelity* halaman master items

Halaman utama pada master *items* sendiri terdiri dari beberapa fitur yang dapat memudahkan admin dalam mengelola *items*, seperti fitur tambah *items*, fitur *export* dan *import* data *items*, *delete*, serta *edit items*.

### b. Perancangan *Unified Model Language* (UML)

Setelah dilakukan tahap perancangan antarmuka, tahap selanjutnya adalah merancang UML yang merupakan suatu *tools* untuk membantu *programmer* dalam membangun sebuah pemodelan sistem yang berorientasi pada objek [16]. Saat ini, UML digunakan sebagai

standarisasi dalam dunia industri untuk memvisualisasikan perancangan sistem [23]. Dalam hal ini, jenis UML yang digunakan untuk merancang sistem peminjaman aset ialah *Use Case Diagram*. Diagram ini menjabarkan setiap proses yang dapat dijalankan oleh masing-masing *role* yang ditunjukkan pada Gambar 4.

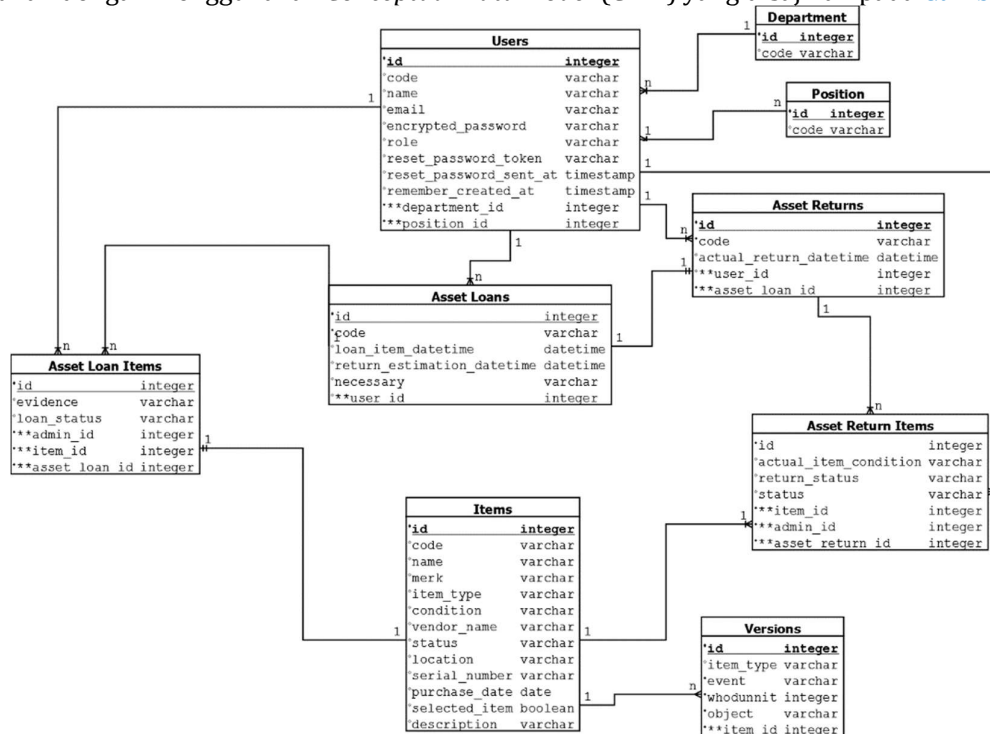


Gambar 4. Perancangan Use Case Diagram

*Use Case Diagram* sistem peminjaman aset ini terdiri dari 2 *role*, yakni *user* dan *admin*. Dalam hal ini, *user* hanya dapat melakukan input data pengajuan peminjaman aset (*asset loans*) dan pengembalian aset (*asset returns*). Sedangkan *admin* dapat melakukan berbagai proses manajemen data, meliputi data *users*, *items*, *department*, *position*, pengelolaan data *asset loans* dan *asset loan items* serta data *asset returns* dan *asset return items*.

c. Perancangan Database

Tahap perancangan terakhir dalam *design workshop* adalah merancang model *database* untuk membangun sistem peminjaman aset. Tahap perancangan model *database* ini akan dilakukan dengan menggunakan *Conceptual Data Model* (CDM) yang disajikan pada Gambar 5.



Gambar 5. Perancangan conceptual data model

Perancangan *database* yang disajikan dalam CDM ini menunjukkan bahwa sistem peminjaman aset Ukirama terdiri dari 9 tabel, yakni tabel *users*, tabel *items*, tabel *departement*, dan *position* yang menyimpan data master, tabel *asset loans*, *asset loans items*, *asset return*, *asset return items* yang menyimpan data transaksi proses peminjaman dan pengembalian aset, dan tabel *versions* yang menyimpan data *history* proses manajemen aset oleh admin. Dalam hal ini, tabel *asset loans* dan *asset returns* menyimpan data peminjaman dan pengembalian secara *general*, sedangkan tabel *asset loan items* dan *asset return items* menyimpan data peminjaman dan pengembalian secara detail, seperti aset apa yang dipinjam, tanggal estimasi pengembalian, serta tanggal pengembalian aktual.

### 2.3 Implementation

Dalam penelitian ini, sistem dibangun dengan bahasa pemrograman *Ruby on Rails* dan *database* PgAdmin. Sistem ini dibangun dengan spesifikasi komputer dengan minimum RAM 8 GB, *processor* Intel core-i5, dan *operating system* Ubuntu Linux. Proses pembangunan sistem dilakukan dengan menginput beberapa *dependency* ke dalam *GemFile* yang merupakan suatu *package* yang dimiliki oleh *Rails* dan digunakan untuk mengeksekusi sistem yang sedang dibangun. *GemFile* tersebut antara lain *Paper Trail* yang digunakan untuk mencatat setiap perubahan yang terjadi pada *migration model*, *Kaminari* untuk membuat *pagination*, *Devise* untuk membuat *migration model users* pada *database* yang dilengkapi dengan autentikasi, dan *PgAdmin* untuk membuat *migration model* ke dalam *database*. Proses implementasi model RAD dalam pembangunan sistem peminjaman aset ini tidak terlepas dari keterlibatan *user* dalam setiap prosesnya. Dalam proses pembangunan sistem, terdapat beberapa *feedback* dan pengembangan fitur yang diberikan oleh *user* kepada *developer* untuk menghasilkan suatu sistem yang bermanfaat dan sesuai dengan *user requirement*. *Feedback* dan pengembangan fitur tersebut antara lain:

Tabel 1. Feedback dan pengembangan fitur sistem peminjaman aset

<b>Pengembangan 1</b>
a. Tidak perlu ada <i>forgot password page</i> , jika <i>user</i> lupa <i>password</i> , dapat langsung menghubungi admin.
b. Tidak perlu ada <i>sign up page</i> . Jika terdapat <i>user</i> baru, maka admin akan menginput data <i>user</i> baru ke dalam sistem.
c. Tidak perlu ada <i>profile page</i> , cukup menginformasikan <i>email user</i> yang sedang <i>login</i> saja di bagian navbar.
d. Ada status pengembalian ( <i>early/ on time/ late</i> ) yang di-setting berdasarkan sistem.
e. Tambahkan <i>page</i> detail data <i>user</i> , detail data <i>item</i> , detail data <i>loan item</i> , detail data <i>return item</i> .
<b>Pengembangan 2</b>
a. Ada 4 status proses pengajuan peminjaman ( <i>waiting, accepted, canceled, dan decline</i> ) yang diinput oleh <i>user</i> .
b. <i>Index</i> peminjaman dan pengembalian <i>item</i> ditampilkan dalam satu <i>index</i> yang sama.
c. Peminjaman dan pengembalian <i>item</i> oleh <i>user</i> dilakukan pada satu <i>page</i> yang sama. <i>User</i> mengajukan peminjaman, jika peminjaman sudah disetujui oleh admin, status peminjaman akan berubah menjadi <i>accepted</i> .
d. <i>Index</i> peminjaman dan pengembalian <i>item</i> pada admin ditampilkan secara terpisah karena admin melakukan dua aksi yang berbeda pada pengajuan peminjaman dan pengembalian <i>item</i> , yakni <i>accepted</i> dan <i>received</i> .
e. Terdapat <i>icon</i> avatar, fitur <i>forgot password</i> dan <i>logout</i> pada <i>dropdown</i> akun <i>user</i> .
<b>Pengembangan 3</b>
a. <i>User</i> dapat meminjam beberapa aset dalam satu waktu.
b. Tambahkan fitur <i>pagination</i> serta <i>multifunctional search &amp; filter data</i> .

- c. Aset yang telah dipinjam akan berubah status dari *available* menjadi *unavailable*.
- d. Admin dapat mengatur kondisi *item* (layak pakai/ layak pakai ada cacat/ tidak layak pakai).
- e. User dapat mengatur kondisi aset (layak pakai/ layak pakai ada cacat/ tidak layak pakai) ketika mengembalikan aset.

#### Pengembangan 4

- a. Admin dapat me-*reset password user* di data master *user*.
- b. Tambahkan fitur *multi fungsional header table* untuk menampilkan data baik secara *ascending* maupun *descending*.
- c. Tambahkan fitur *import* dan *export data to csv*.
- d. Yang dapat mengatur kondisi *item* (sebelum dan sesudah dipinjam) hanya admin.

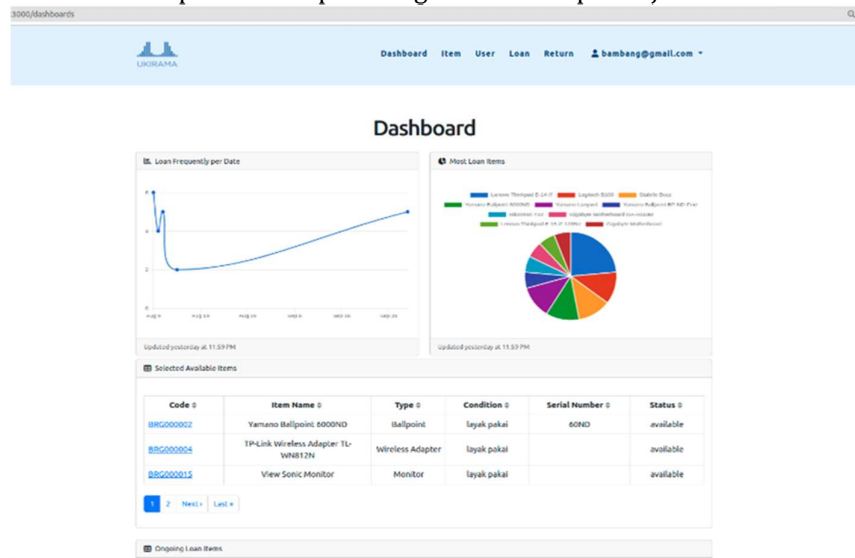
#### Pengembangan 5

- a. Ketika melakukan *action* pada peminjaman maupun pengembalian *item*, admin harus menginputkan siapa admin yang menyetujui/menerima ataupun menolak pengajuan peminjaman disertai dengan alasannya.
- b. Tambahkan fitur *history* sebagai log *action* yang telah dilakukan oleh admin dalam melakukan pengelolaan data.
- c. Tambahkan fitur *dashboard* untuk admin pada halaman ini akan ditampilkan data *item* yang paling sering dipinjam, frekuensi peminjaman *item* per hari, data *item* pilihan yang *available*, data *item* yang sedang dipinjam, dan data *record user* yang paling sering terlambat mengembalikan *item*.

Mengacu pada **Tabel 2**, dapat dilihat proses perancangan sistem peminjaman aset ini dilakukan dengan melewati lima kali tahap pengembangan sistem dimana proses pengembangan ini tidak dapat terlepas dari keterlibatan *user*. Setelah proses perancangan selesai, tahap selanjutnya adalah implementasi pembangunan sistem yang dilakukan dengan menggunakan bahasa pemrograman *Ruby on Rails*.

### 3. HASIL DAN PEMBAHASAN

Tahap perancangan dan pemodelan yang telah dilakukan digunakan sebagai acuan untuk membangun sistem peminjaman aset. Proses rancang bangun sistem peminjaman aset yang dilakukan dengan menerapkan model RAD ini berlangsung selama  $\pm 60$  hari kerja dengan pengerjaan 5 jam/hari. Saat ini, sistem peminjaman aset ini digunakan secara lokal dalam perusahaan. Berikut merupakan hasil pembangunan sistem peminjaman aset Ukirama.



Gambar 6. Halaman dashboard admin



**Gambar 6** menunjukkan hasil tampilan *dashboard admin*. *Dashboard* ini menyajikan beberapa informasi mengenai aset yang paling sering dipinjam, frekuensi peminjaman aset setiap harinya, data keterangan aset pilihan, serta data pengembalian aset yang diurutkan mulai dari yang terlambat hingga yang paling tepat waktu.

Code	Name	Type	Vendor Name	Condition	Location	Serial Number	Purchase Date	Status
BIG000001	Yamano Ballpoint BP-ND-Fine	Ballpoint	Yamano	Good	office	BPN342D	2023-05-10	available
BIG000002	Yamano Ballpoint 6000ND	Ballpoint	Yamano	Good	office	60ND	2023-02-16	available
BIG000003	TP-Link Wireless Adapter TL-WN812N	Wireless Adapter	TP-Link	Good	office		2023-08-05	available
BIG000004	TP-Link Wireless Adapter TL-WN812N	Wireless Adapter	TP-Link	Good	office		2023-06-09	available
BIG000005	Lenovo Thinkpad E-14 I7	Laptop	Lenovo	Good	office		2022-04-21	available
BIG000006	Logitech K120	Keyboard	Logitech	Good	office	K120	2022-02-10	available
BIG000007	Gigabyte Motherboard	Motherboard	Gigabyte	Good	office		2023-01-04	available
BIG000008	Lenovo Thinkpad E-15 I7-1255U	Laptop	Lenovo	Good	office	1235U	2022-11-09	available
BIG000009	Logitech K120	Keyboard	Logitech	Good	office	K120	2020-08-06	available
BIG000010	Logitech B100	Mouse	Logitech	Good	office		2021-08-20	available

**Gambar 7.** Halaman index master items-admin

Halaman index master *items* yang ditunjukkan pada **Gambar 7** menyajikan beberapa informasi mengenai data kelengkapan *items* yang dapat diakses oleh admin. Dalam hal ini, *items* yang disajikan pada index *items* bersifat *confidential*. Halaman ini juga menyediakan fitur import dan ekspor yang digunakan oleh admin untuk mengelola data.

Setelah sistem dibangun, selanjutnya sistem akan diuji menggunakan metode *Blackbox Testing* dengan skenario pengujian validitas sistem yang disajikan pada **Tabel 2**.

**Tabel 2.** Hasil pengujian sistem menggunakan *Blackbox Testing*

No.	Skenario Pengujian	Hasil Diharapkan	Hasil Pengujian	Kesimpulan
1.	Admin atau <i>User</i> melakukan <i>login</i> sistem dengan mengisi <i>username</i> dan <i>password</i> .	Admin atau <i>User</i> berhasil <i>login</i> .	Admin atau <i>User</i> berhasil <i>login</i> . Jika <i>username</i> atau <i>password</i> yang diisi benar, muncul pesan berhasil <i>login</i> . Jika <i>username</i> atau <i>password</i> yang diisi salah, muncul pesan gagal <i>login</i> dan diarahkan kembali ke halaman <i>login</i> .	Valid
2.	Admin dapat menambah/ menyunting master data <i>item</i> dan <i>user</i> .	Data <i>item</i> dan <i>user</i> baru tersimpan dalam <i>database</i> .	Data <i>item</i> baru berhasil tersimpan ke dalam <i>database</i> , muncul pesan "Berhasil menambahkan/mengubah <i>item/user</i> ".	Valid
3.	Admin dapat menghapus data master <i>item</i> .	Data terhapus dari table <i>item</i> .	Data pada table <i>item</i> terhapus, tetapi proses menghapus data ter- <i>record</i> ke dalam fitur <i>history</i>	Valid

<i>items.</i>				
4.	Admin dapat mengatur kondisi <i>item</i> (layak pakai/ layak pakai ada cacat/ tidak layak pakai).	Data kondisi <i>item</i> tersimpan dalam <i>database</i> .	Data kondisi <i>item</i> tersimpan dalam <i>database</i> dan ditampilkan dalam bentuk <i>pill badges</i> .	Valid
5.	Admin dapat mengatur status <i>item</i> ( <i>available/unavailable</i> ) secara manual.	Data perubahan status <i>item</i> tersimpan dalam <i>database</i> .	Data perubahan status <i>item</i> tersimpan dalam <i>database</i> .	Valid
6.	Admin dapat menambahkan beberapa foto dokumentasi <i>item</i> .	Data foto dokumentasi <i>item</i> tersimpan ke dalam <i>database</i> .	Data foto dokumentasi <i>item</i> tersimpan ke dalam <i>database</i> berupa URL.	Valid
7.	Admin dapat mengimpor/mengekspor data master <i>items</i> .	Data master <i>items</i> berhasil diimpor/diekspor.	Data master <i>items</i> dapat diimpor/diekspor ke dalam bentuk CSV.	Valid
8.	Admin dapat mereset <i>password</i> user jika user lupa <i>password</i> .	Data <i>password user</i> berhasil ter-reset.	Data <i>password user</i> berhasil ter-reset, user dapat masuk ke sistem peminjaman dengan <i>password</i> yang baru.	Valid
9.	Admin dapat mengekspor data master <i>users/loans/returns</i> .	Data master <i>users/loans/returns</i> berhasil di ekspor.	Data master <i>users/loans/returns</i> diekspor ke dalam bentuk CSV.	Valid
10.	Admin dapat melakukan pencarian data berdasarkan semua nama <i>column</i> pada halaman index master <i>items</i> , master <i>users</i> , <i>loans</i> , dan <i>returns</i> .	Data yang dicari berhasil ditemukan dan difilter.	Data yang dicari berhasil ditemukan dan di filter dengan menggunakan <i>multifunctional search</i> .	Valid
11.	Admin dapat menerima atau menolak pengajuan <i>loan item</i> .	Admin dapat memberikan <i>action</i> pada pengajuan <i>loan item</i> .	Admin dapat memberikan <i>action accepted/rejected/canceled</i> pada pengajuan <i>loan item</i> . Jika <i>user</i> sudah melakukan pengajuan, tetapi admin belum memberikan <i>action</i> , maka status pengajuan peminjaman menjadi " <i>waiting</i> ".	Valid
12.	Terdapat status <i>available/unavailable</i> pada <i>item</i> .	Status <i>available/unavailable</i> terjadi pada	Ketika <i>user</i> melakukan pengajuan peminjaman <i>item</i>	Valid

		saat pengajuan loan item.	dan disetujui oleh admin, maka status <i>item</i> akan berubah menjadi <i>unavailable</i> .	
13.	Sistem me- <i>record</i> semua <i>action</i> yang dilakukan oleh admin pada pengajuan <i>loan item</i> .	Semua <i>action</i> yang dilakukan oleh admin pada pengajuan <i>loan item</i> , ter- <i>record</i> ke dalam <i>log</i> .	Semua <i>action</i> yang dilakukan oleh admin pada pengajuan <i>loan item</i> ( <i>accepted/ rejected/ canceled</i> ) ter- <i>record</i> ke dalam fitur <i>history</i> .	Valid
14.	Admin dapat menerima pengajuan pengembalian <i>item</i> .	Admin dapat menerima pengajuan pengembalian <i>item</i> dengan memberikan <i>action</i> "received".	Admin dapat menerima pengajuan pengembalian <i>item</i> dengan memberikan <i>action</i> "received".	Valid
15.	Terdapat status ketepatan waktu pengembalian <i>item</i> .	Sistem secara otomatis mencatat ketepatan waktu pengembalian <i>item</i> yang dilakukan oleh <i>user/ admin</i> .	Ketika <i>user</i> mengembalikan <i>item</i> , akan muncul status pengembalian ( <i>early/ on time/ late</i> ) pada sistem secara otomatis.	Valid
16.	Sistem me- <i>record</i> semua <i>action</i> yang dilakukan oleh admin pada pengajuan <i>loan item</i> .	Semua <i>action</i> yang dilakukan oleh admin pada pengajuan <i>loan item</i> , ter- <i>record</i> ke dalam <i>log</i> .	Semua <i>action</i> yang dilakukan oleh admin pada pengajuan <i>loan item</i> ( <i>accepted/ rejected/ canceled</i> ) ter- <i>record</i> ke dalam fitur <i>history</i> .	Valid
17.	Admin dapat mengajukan peminjaman atau pengembalian untuk dirinya sendiri maupun untuk <i>user</i> lain.	Admin dapat memproses peminjaman atau pengembalian secara mandiri.	Admin dapat memberikan <i>action</i> "accepted/ decline/ canceled" secara mandiri pada pengajuan <i>loan item</i> dan <i>action</i> "received" pada pengajuan <i>return item</i> .	Valid
18.	<i>User</i> dapat mengajukan peminjaman lebih dari 1 <i>items</i> .	<i>User</i> dapat mengajukan peminjaman pada <i>items</i> yang <i>available</i> saja.	<i>User</i> dapat mengajukan peminjaman pada <i>items</i> yang <i>available</i> saja. Hanya <i>items</i> yang <i>available</i> yang dimunculkan pada form peminjaman <i>user</i> .	Valid

19.	<i>User</i> hanya dapat melihat <i>items</i> yang dipinjam dan dikembalikan oleh dirinya sendiri.	<i>Dashboard user</i> hanya menampilkan data <i>loan</i> dan <i>return items</i> oleh masing-masing <i>user</i> .	<i>Dashboard user</i> menampilkan data <i>loan</i> dan <i>return items</i> dari masing-masing <i>user</i> yang dilengkapi dengan tanggal peminjaman, tanggal estimasi pengembalian, dan tanggal pengembalian <i>items</i> aktual.	Valid
20.	<i>User</i> dapat membatalkan pengajuan peminjaman.	<i>User</i> dapat membatalkan pengajuan peminjaman <i>items</i> selama pengajuan belum disetujui oleh admin.	<i>User</i> dapat menghapus pengajuan peminjaman <i>items</i> dari akunnya sendiri atau mengajukan <i>cancel</i> secara langsung pada admin.	Valid

Hasil uji menggunakan metode *blackbox testing*, menunjukkan bahwa sistem peminjaman aset ini 100% berhasil dan sesuai dengan *requirement* yang telah ditentukan di awal bersama *user*.

#### 4. SIMPULAN

Dari penelitian yang telah dilakukan pada PT Ukirama Solusi Indonesia, diperoleh hasil bahwa rancang bangun sistem peminjaman aset menggunakan model RAD dapat mempercepat tahap pembangunan sistem dimana proses rancang bangun sistem berlangsung selama ± 60 hari kerja dengan pengerjaan 5 jam/hari. Selain itu, penerapan model RAD ini dapat menghasilkan sistem dengan fungsionalitas yang baik dan sesuai dengan *user requirement* dimana hasil *Blackbox Testing* menunjukkan *feedback* 100% berhasil. Untuk masa yang akan mendatang, peneliti berikutnya dapat mengembangkan sistem peminjaman aset dari sisi kompleksitas *requirement* yang diperoleh dari *user*, misalnya dikembangkan menjadi sistem penyewaan aset.

#### REFERENSI

- [1] P. P. K. Wardani, "Produktivitas Kerja Karyawan Pada Pt .Polymatech Indonesia," *J. Ekbis*, vol. 8, no. 1, 2020.
- [2] Dani Yusuf, "SISTEM PEMINJAMAN BARANG DI PERUSAHAAN MENGGUNAKAN TEKNOLOGI RFID," *J. Teknol. Pelita Bangsa*, vol. 8, no. 1, pp. 43–52, 2017.
- [3] T. Pricillia and Zulfachmi, "Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)," *J. Bangkit Indones.*, vol. 10, no. 1, pp. 6–12, 2021, doi: 10.52771/bangkitindonesia.v10i1.153.
- [4] A. Puspaningrum and E. Sudarmilah, "SISTEM INFORMASI MANAJEMEN PEMINJAMAN ( STUDI KASUS : PENGELOLAAN ASET DAN TATA RUANG TAMAN BUDAYA JAWA TENGAH )," *Technologia*, vol. 11, no. 1, pp. 37–45, 2020.
- [5] M. I. Ramadhan and B. D. Andah, "Implementasi Sistem Informasi Peminjaman Aset Kampus Berbasis Web Pada Direktorat Administrasi Umum Universitas Budi Luhur Dengan Metodologi Berorientasi Obyek," *Idealis*, vol. 1, pp. 334–339, 2018.
- [6] G. Gumilar and R. Wulan, "Sistem Informasi Peminjaman Aset Barang Kantor Di Lapas Kelas II A Cibinong," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 5, pp. 819–825, 2022, doi: 10.32672/jnkti.v5i5.4821.

- [7] A. M. Callysta and M. A. I. Pakereng, "Analisis Dan Perancangan Sistem Inventarisasi Aset Pada Pt. Bangunindo Teknusa Jaya," *J. Tek. Inf. dan Komput.*, vol. 5, no. 1, p. 19, 2022, doi: 10.37600/tekinkom.v5i1.499.
- [8] S. Dewi, L. M. Jannah, and Y. Jumaryadi, "Analisis dan Perancangan Sistem Informasi Manajemen Aset Tetap pada PT. Metis Teknologi Corporindo," *J. Sist. Informasi, Teknol. Inform. dan Komput.*, vol. 9, no. September, pp. 81–91, 2018.
- [9] Darwin and N. E. Budiyanata, "Rancang Bangun Sistem Peminjaman dan Manajemen Aset Laboratorium Berbasis Implementasi RFID dan Aplikasi Web," *J. Edukasi Elektro*, vol. 05, no. 2, pp. 80–90, 2021.
- [10] R. Parlika, M. Afifudin, I. A. Pradana, Y. D. W. Wiratama, and M. N. Holis, "Studi Literatur Efisiensi Model Rapid Application Development dalam Pengembangan Perangkat Lunak (2014-2022)," *Positif J. Sist. dan Teknol. Inf.*, vol. 8, no. 2, pp. 64–73, 2023, [Online]. Available: <http://mcastud.com/student-project-development-go/>.
- [11] M. Muhasshanah and S. Qamariyah, "Implementasi Sistem Informasi Manajemen Laboratorium Kebidanan Pada Fakultas Ilmu Kesehatan Universitas Ibrahimy," *J. Ilm. Inform.*, vol. 6, no. 2, pp. 81–92, 2021, doi: 10.35316/jimi.v6i2.1234.
- [12] I. Dewi Sintawati, "Komparasi Metode Rad Dengan Rup Pada Pengembangan Sistem Informasi," *Akrab Juara J. Ilmu-ilmu Sos.*, vol. 7, no. 2, p. 101, 2022, doi: 10.58487/akrabjuara.v7i2.1796.
- [13] M. P. Putri and H. Effendi, "Implementasi Metode Rapid Application Development Pada Website Service Guide 'Waterfall Tour South Sumatera,'" *J. SISFOKOM*, vol. 07, no. 02, pp. 130–136, 2018.
- [14] R. Yusuf, S. Muharni, and R. Hasbid, "Penerapan Waterfall Model Pada Perancangan Sistem Pelayanan Dan Informasi Dengan Pendekatan Ooad Menggunakan Uml," *Int. Res. Big-Data Comput. Technol. I-Robot*, vol. 5, no. 1, pp. 1–6, 2021, doi: 10.53514/ir.v5i1.176.
- [15] R. Sukmawati and Y. Priyadi, "Perancangan Proses Bisnis Menggunakan UML Berdasarkan Fit/Gap Analysis Pada Modul Inventory Odoo," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 3, no. 2, p. 104, 2019, doi: 10.29407/intensif.v3i2.12697.
- [16] J. Margaretha and A. Voutama, "Perancangan Sistem Informasi Pemesanan Tiket Konser Musik Berbasis Web Menggunakan Unified Modeling Language (UML)," *JOINS (Journal Inf. Syst.*, vol. 8, no. 1, pp. 20–31, 2023, doi: 10.33633/joins.v8i1.7107.
- [17] I. M. S. Ardana and Y. M. Djaksana, "PERANCANGAN BASIS DATA KAWASAN SUCI DANAU TAMBLINGAN DENGAN MENERAPKAN MODEL DATA RELASIONAL," *J. Syntax Dmiration*, vol. 4, no. 10, pp. 1598–1612, 2023.
- [18] M. Lailiya, N. L. W. S. R. Ginantra, and G. S. Mahendra, "Journal homepage: <https://journal.literasisains.id/index.php/jomlai/> Sistem Informasi Penyesuaian Anggaran Belanja Berbasis Website Pada PT. Taspen (Persero) Kantor Cabang Denpasar," *JOMLAI J. Mach. Learn. Artif. Intell.*, vol. 1, no. 1, pp. 31–42, 2022, doi: 10.55123/jomlai.v1i1.162.
- [19] A. Fahrezi, F. N. Salam, G. M. Ibrahim, R. R. Syaiful, and A. Saifudin, "Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT . AINO Indonesia," *J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 1, pp. 1–5, 2022.
- [20] N. M. D. Febriyanti, A. A. K. O. Sudana, and I. N. Piarsa, "Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen," *J. Ilm. Teknol. dan Komput.*, vol. 2, no. 3, 2021.
- [21] M. R. Rifaldi and A. Imamuddin, "Rancang analisis aplikasi software sistem pembayaran koperasi menggunakan metode prototyping," *J. Inform. Teknol.*, vol. 3, no. 2, pp. 61–70, 2022.
- [22] Supiyandi, C. Rizal, and B. Fachri, "Implementasi Model Prototyping Dalam Perancangan Sistem Informasi Desa," *RESOLUSI Rekayasa Tek. Inform. dan Inf.*, vol. 3, no. 3, pp. 211–216, 2023.

- [23] M. A. Sobarnas and I. Sofiya, "Analisis dan perancangan sistem informasi pengolahan data rekam medis berbasis WEB di puskesmas XYZ," *J. Inform.*, vol. 3, no. 2, pp. 82-91, 2022, doi: 10.37373/infotech.v3i2.350.