

Pengelolaan Keuangan Pribadi Menggunakan Teknologi Citra Digital Berbasis Website.pdf

by

Submission date: 17-Apr-2023 07:30AM (UTC+0700)

Submission ID: 2066396920

File name: Pengelolaan Keuangan Pribadi Menggunakan Teknologi Citra Digital Berbasis Website.pdf
(1.07M)

Word count: 4169

Character count: 26690



@is The Best :
Accounting Information Systems and
Information Technology Business Enterprise
Volume 7, Nomor 2 (2022) Hal. 136-150
ISSN: 2252-9853 (Print) | ISSN: 2656-808X (Online)
<https://ojs.unikom.ac.id/index.php/aisthebest/index>

Terakreditasi Peringkat 4, SK No.: 105/E/KPT/2022

DOI: <https://doi.org/10.34010/aisthebest.v7i2.8957>

Pengelolaan Keuangan Pribadi Menggunakan Teknologi Citra Digital Berbasis *Website*

Heri Purwanto¹, Rikky Wisnu Nugraha², Deden Sofyan Hamdani³, Mohamad Rizky⁴

¹Universitas Sangga Buana, Jl. Phh. Mustofa No.68 Bandung 40124 Jawa Barat-Indonesia

^{2,3,4}Institut Digital Ekonomi LPKIA, Jl. Soekarno Hatta 456 Bandung 40266 Jawa Barat-Indonesia

Email: heri.purwanto@usbykpk.ac.id

ABSTRACT

Financial management is an action to meet future financial needs and can also be used to solve personal and group economic problems. Financial management is needed to deal with financial problems that occur in life. Increasing daily expenses are one of the reasons for the importance of managing personal finances so you don't get stuck in debt. The community is also not used to managing their finances, so they are often burdened with unexpected expenses. By managing the right finances, it can be useful to help the implementation of each individual goal. This study aims to develop applications that have been previously designed by previous researchers, the system is made on a website-based basis for user flexibility, and features are added using digital image technology to add input features for financial income and expenditure data in the form of money scans and shopping receipt scans. The methodology used in this research is the Agile method, this methodology focuses on customer satisfaction and also the speed of program development. The data collection technique uses literature study and the modeling language used is the Unified Modeling Language (UML). Based on the test results, this system can be accessed through various operating systems that have an internet connection and a browser, making it easier for users to take advantage of the various facilities in the system. This system can be developed further by adding features that can remind users of the use of money, such as notifications about limits on financial use.

Keywords: Money, Financial management, Recording, UML

ABSTRAK

Pengelolaan keuangan merupakan suatu tindakan untuk mencukupi kebutuhan keuangan pada masa mendatang dan juga dapat digunakan untuk menyelesaikan masalah ekonomi pribadi maupun kelompok. Pengelolaan keuangan dibutuhkan untuk menangani persoalan keuangan yang terjadi dalam kehidupan. Pengeluaran biaya sehari-hari yang semakin bertambah, menjadi satu dari beberapa sebab pentingnya mengatur keuangan pribadi agar tidak terjebak dalam hutang. Masyarakat juga belum terbiasa dalam mengelola keuangannya, sehingga sering terbebani dengan pengeluaran yang tidak terduga. Dengan mengelola keuangan yang tepat, dapat berguna untuk membantu terlaksananya setiap tujuan individu. Penelitian ini bertujuan untuk mengembangkan aplikasi yang telah dirancang sebelumnya oleh peneliti terdahulu, sistem dibuat dengan berbasis *website* demi fleksibilitas pengguna, dan ditambahkan fitur menggunakan teknologi citra digital menambahkan fitur *input* data pemasukan dan pengeluaran keuangan berupa *scan* uang dan *scan* struk belanja. Metodologi yang digunakan dalam penelitian ini adalah metode *Agile*, metodologi ini berfokus kepada kepuasan pelanggan dan juga kecepatan pembuatan program. Teknik pengumpulan data menggunakan studi pustaka dan bahasa pemodelan yang digunakan adalah *Unified Modelling Language* (UML). Berdasarkan hasil pengujian, sistem ini mampu diakses melalui berbagai sistem operasi yang memiliki koneksi internet dan *browser*, sehingga memudahkan pengguna dalam memanfaatkan berbagai fasilitas yang ada dalam sistem. Sistem ini dapat dikembangkan lebih jauh dengan menambahkan fitur yang dapat mengingatkan pengguna dalam penggunaan uang, seperti notifikasi limit penggunaan keuangan.

Kata Kunci: Uang, Manajemen Keuangan, Pencatatan, UML

Pendahuluan

Pengelolaan keuangan sangatlah dibutuhkan untuk menangani masalah keuangan yang timbul dalam kehidupan umum. Peneliti lain menyebutkan bahwa, mengatur keuangan adalah kunci untuk

mencapai kesuksesan finansial. Kebutuhan hidup yang terus bertambah, menjadikan salah satu sebab pentingnya mengatur keuangan agar tidak terjebak dalam hutang [1]. Mengelola keuangan yang benar sangat berperan untuk membantu meraih tiap target orang yang diidamkan. Agar mampu mengelola keuangan dengan benar, maka setiap orang perlu melakukan pengelolaan keuangan yang mencakup uang masuk dan uang keluar [2]. Berdasarkan hal tersebut, membuktikan bahwa pengelolaan keuangan merupakan hal penting dilakukan seseorang agar dapat mengetahui seluruh pemasukan dan pengeluaran uang.

Generasi saat ini masih banyak yang malas atau terlalu sibuk sehingga lalai dalam mencatat berbagai pengeluaran keuangan yang dilakukan sehari-hari [5][10], sebagian lainnya merasa *stress* karena tidak dapat mengatur pengelolaan keuangan [2]–[11], hal tersebut berdampak pada lemahnya kemampuan dalam merencanakan dan mengendalikan penggunaan keuangan untuk pencapaian tujuan [4]. Lemahnya pengelolaan keuangan tersebut, berakibat pada pengeluaran lebih besar dari pendapatan [8]. Sebaliknya jika setiap pribadi dapat mengelola keuangan dengan optimal, maka keuangan yang dimilikinya berpotensi untuk investasi masa depan [2]–[11].

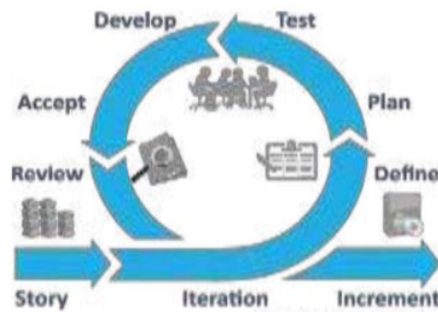
Berdasarkan hasil analisis dari 12 (dua belas) artikel ilmiah yang paling berkaitan dengan tema penelitian ini, terdapat 10 (sepuluh) peneliti membuat aplikasi serupa dengan berbasis android [2]–[11], beberapa diantaranya dilengkapi dengan fasilitas seperti *speech to Text* [3] dan grafis [9]. Terdapat 1 (satu) peneliti yang membuat aplikasi serupa dengan berbasis *desktop* [12], dan 1 (satu) peneliti lainnya berbasis *chatbot* [13]. Aplikasi yang telah dibuat memiliki fitur antara lain: *input* data pemasukan dan pengeluaran dengan cara mengetik manual, laporan keuangan berdasarkan waktu (hari, bulan, tahun), infografis atau grafik persentase data, dan *Speech to Text* untuk *input* data. Dari beberapa fitur tersebut, dapat terlihat bahwa *input* data keuangan masih dilakukan secara manual, sehingga akan terjadi kemungkinan kesalahan. Sejalan dengan penelitian sebelumnya, pada penelitian ini akan menambahkan fitur *input* data pemasukan keuangan dengan memanfaatkan teknologi citra digital berupa *scan* uang, dan fitur *input* data pengeluaran keuangan menggunakan *scan* struk belanja. Selain hal tersebut, pada penelitian ini aplikasi dikembangkan berbasis *web*, dimana hal tersebut belum dilakukan oleh penelitian terdahulu.

Penggunaan teknologi citra digital sendiri sebenarnya telah banyak dimanfaatkan di berbagai bidang [14], tidak terkecuali dalam hal pengelolaan keuangan pribadi dimana komputer mampu membaca uang kertas/struk belanja dengan menggunakan kamera (*webcam*) atau melalui *scan* gambar dua dimensi [15] yang diunggah, sehingga selain *input* data dapat dilakukan secara cepat, juga kekeliruan dalam mengidentifikasi uang kertas/struk belanja dapat dihindari [16]. Penggunaan fitur yang didukung teknologi citra digital ini dapat dimanfaatkan untuk mempercepat dan mempermudah dalam menggunakan aplikasi ini. Penelitian ini bertujuan untuk mengembangkan aplikasi yang telah dirancang sebelumnya oleh peneliti terdahulu, sistem dibuat dengan berbasis *website* demi fleksibilitas pengguna dan penggunaan yang lebih luas dengan berbagai pilihan sistem operasi yang digunakan dalam komputer, dan ditambahkan fitur *scan* gambar untuk memudahkan input data pemasukan dan pengeluaran.

Metode

Metode pengembangan sistem yang digunakan pada penelitian ini adalah *Agile Software Development*. *Agile Software Development* adalah cara gabungan proses atau prosedur pembangunan perangkat lunak yang berlandaskan pada pembangunan iteratif, dimana ketentuan dan solusi berkembang dengan kerja sama antar rekan kerja yang tertata, fokus pada perkembangan yang cepat, dan membuat kode bernilai tinggi serta pada prosedur pertumbuhannya mengaitkan konsumen secara langsung [17]. Peneliti lain menyatakan bahwa *Agile* adalah metode manajemen proyek yang dibuat dengan siklus pembangunan yang ringkas untuk fokus pada kenaikan berkepanjangan dalam pembangunan suatu produk atau pelayanan. Pada proses pembangunannya,

agile bisa disebut proses pembangunan yang singkat karena proses sentral dari pengembangan *agile* berpusat pada *design-code-test once day* [18]. Adapun langkah-langkah *agile* yang dilakukan sebagai berikut: 1) mengatur rincian prioritas bagi fitur-fitur yang akan dibangun pada aplikasi ini, 2) dilanjutkan dengan mengatur kegiatan yang akan diperbuat untuk memenuhi kebutuhan yang ditentukan dalam *backlog* dengan durasi realisasi selama 30 hari kerja, 3) mengadakan rapat dengan rekan kerja yang sudah dipilih untuk menelaah kemajuan kegiatan dalam pengembangan aplikasi manajemen keuangan pribadi, dan 4) menampilkan fitur-fitur perangkat lunak yang telah dibuat agar dapat dilakukan penilaian oleh pengguna sesuai dengan waktu yang telah ditetapkan [17]. Selain menggunakan metode *Agile*, pembuatan sistem ini juga menggunakan teknik Studi Pustaka dalam pengumpulan data yang relevan dengan judul penelitian ini didapat melalui jurnal, artikel ilmiah, tulisan pada internet atau pada media lainnya. Literatur yang digunakan adalah literatur yang relevan dengan aplikasi metode pemrograman dan seluruh informasi yang berhubungan dengan aplikasi yang hendak dirancang [6]. Bahasa pemodelan dalam pembuatan aplikasi ini adalah *Unified Modeling Language* (UML) [6].



Gambar 1. Model *Agile* [18]




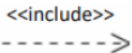
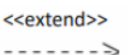


Hasil dan Pembahasan

Aplikasi yang dikembangkan memiliki kegunaan untuk menggabungkan data dari semua sumber dan menggunakan beragam media untuk memunculkan informasi [19]. Manajemen menjadi sebuah proses untuk merencanakan, mengorganisasikan, menggerakkan, dan mengawasi, yang dilaksanakan untuk menetapkan serta meraih target yang sudah ditentukan melalui penggunaan kemampuan manusia untuk meraih target yang telah ditentukan [20]. Pengembangan aplikasi ini lebih menekankan pada pengelolaan keuangan pribadi adalah di mana dengan sistem ini dapat membantu seseorang dalam mempelajari dan mengelola uang [13]. Sistem yang dibangun berbasis *website*, di mana di dalamnya terdapat gabungan laman yang berfungsi untuk memunculkan informasi, skrip, sketsa, animasi dan bunyi. Membentuk satu susunan yang saling berkaitan dan disambungkan pada jaringan-jaringan halaman [21].

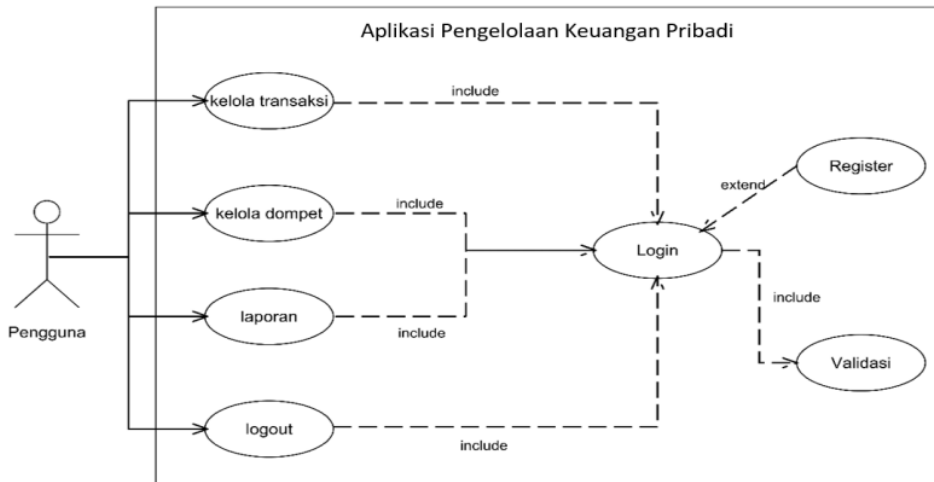
Seperti telah dijelaskan sebelumnya bahwa sistem yang dikembangkan merupakan penyempurnaan dari sistem yang telah dibuat oleh peneliti terdahulu, di mana pada pengembangan sistem terdahulu terdapat beberapa kelemahan, di antaranya: 1) sistem berbasis *desktop* dan *android*, belum dikembangkan ke *platform website*, 2) belum memiliki fitur pengelompokkan uang/dompet, 3) belum memiliki fitur untuk melakukan *scan* gambar/kamera (*webcam*).

Untuk menghasilkan sistem yang baik, maka diperlukan rancangan sistem yang berisi langkah-langkah operasi dalam proses pengolahan data [22][24]. Selanjutnya dalam perancangan ini, pemodelan dilakukan pertama kali dengan menggunakan *usecase diagram*. *Usecase diagram* mengilustrasikan representasi dari interaksi untuk mengetahui hubungan aktor dengan sistem yang dipakai untuk mengetahui kegunaan yang terdapat dalam sebuah sistem [6]. *Usecase diagram* menggunakan beberapa simbol sebagai berikut:

Tabel 1. Simbol *Usecase Diagram* [23]

| No | Simbol | Nama | Keterangan |
|----|---|--------------------|---|
| 1. |  | <i>Actor</i> | Aktor dapat berupa proses, orang, atau sistem lain yang berinteraksi dengan sistem yang dikembangkan. |
| 2. |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen <i>independent</i> akan berpengaruh terhadap elemen <i>dependent</i> . |
| 3. |  | <i>Association</i> | Menunjukkan keterhubungan komunikasi antara aktor dengan <i>usecase</i> . |
| 4. |  | <i>Include</i> | Relasi yang menunjukkan bahwa <i>usecase</i> tambahan yang ditambahkan ke sebuah <i>usecase</i> membutuhkan <i>usecase</i> ini untuk menjalankan fungsinya. |
| 5. |  | <i>Extend</i> | Relasi yang menunjukkan bahwa <i>usecase</i> tambahan yang ditambahkan ke sebuah <i>usecase</i> , dapat berdiri sendiri. |
| 6. |  | <i>System</i> | Menggambarkan sistem yang sedang dibangun. |
| 7. |  | <i>Usecase</i> | Bagian tingkat tinggi dari fungsionalitas yang disediakan oleh sistem. |

Secara keseluruhan fungsionalitas sistem dapat digambarkan pada *usecase diagram* berikut ini:





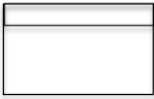





Gambar 2. *Usecase diagram* Aplikasi Pengelolaan Keuangan Pribadi

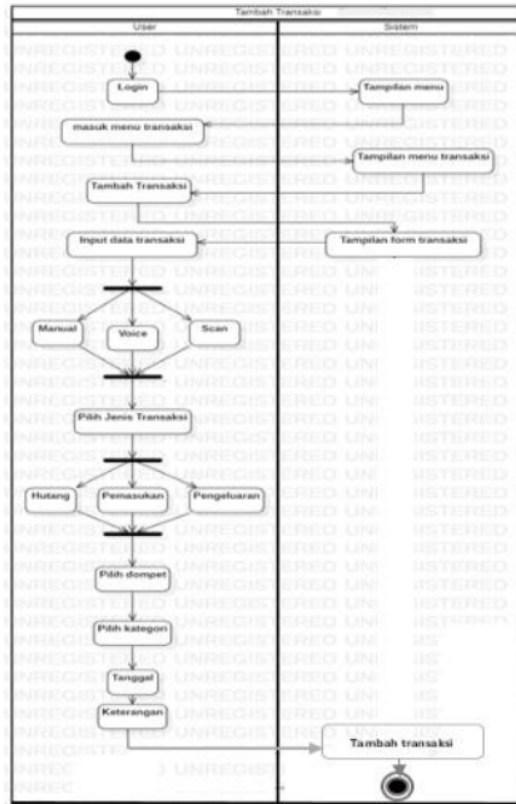
Pada gambar 2 di atas terdapat gambaran fungsionalitas sistem yang dibuat ke dalam bentuk *usecase diagram*. Pengguna dapat *register*, *login*, mengelola transaksi, mengelola dompet, melihat laporan (harian, mingguan, bulanan, tahunan), dan juga *logout* dari sistem.

² Untuk alur langkah-langkah aktivitas setiap *usecase* digambarkan dalam *activity diagram*. *Activity Diagram* adalah diagram spesifik dimana sebagian besarnya adalah *action*. Oleh karena itu, *Activity Diagram* lebih mengilustrasikan langkah-langkah kegiatan dari level atas [23]. Beberapa simbol *activity diagram* yang digunakan disajikan pada tabel 2 berikut ini:

Tabel 2. Simbol *Activity Diagram* [23]

| No | Simbol | Nama | Keterangan |
|----|---|-----------------|--|
| 1. |  | Status awal | Menyatakan status awal dari sebuah aktivitas dalam sebuah sistem, setiap <i>activity diagram</i> memiliki status awal. |
| 2. |  | Aktivitas | Aktivitas yang dilakukan oleh sebuah sistem (biasanya menggunakan kata kerja). |
| 3. |  | <i>Decision</i> | Menunjukkan terjadinya percabangan dimana pilihan aktivitas lebih dari satu. |
| 4. |  | Penggabungan | Menggambarkan lebih dari satu aktivitas digabungkan menjadi satu. |
| 5. |  | <i>Swimlane</i> | Untuk mengelompokkan aktivitas berdasarkan aktor, sehingga organisasi bisnis yang bertanggungjawab terhadap aktivitas dapat terlihat dengan jelas. |
| 6. |  | <i>Fork</i> | Menunjukkan aktivitas dilakukan secara parallel. |
| 7. |  | <i>Join</i> | Menunjukkan aktivitas yang digabungkan. |
| 8. |  | Status akhir | Merupakan status akhir yang dilakukan oleh sistem. |

Seluruh *usecase* yang terdapat pada *usecase diagram*, digambarkan secara keseluruhan dalam *activity diagram*, dari mulai aktivitas *login*, registrasi, tambah dompet, tambah transaksi, sampai dengan *logout* dari sistem. Gambar 2 di bawah ini menunjukkan alur *activity diagram* menambahkan transaksi. Jika pengguna ingin menambahkan transaksi pengguna harus mengisi form yang terdiri dari jenis transaksi, dompet yang akan dipakai, kategori transaksi, tanggal transaksi dan juga keterangan. Apabila pengguna telah mengisi form, data yang telah diisi akan langsung tersimpan ke menu transaksi dalam *database*. Berikut ini *activity diagram* dari sistem yang dibangun tersaji pada gambar 3 berikut ini.






Gambar 3. Activity Diagram: Tambah Transaksi

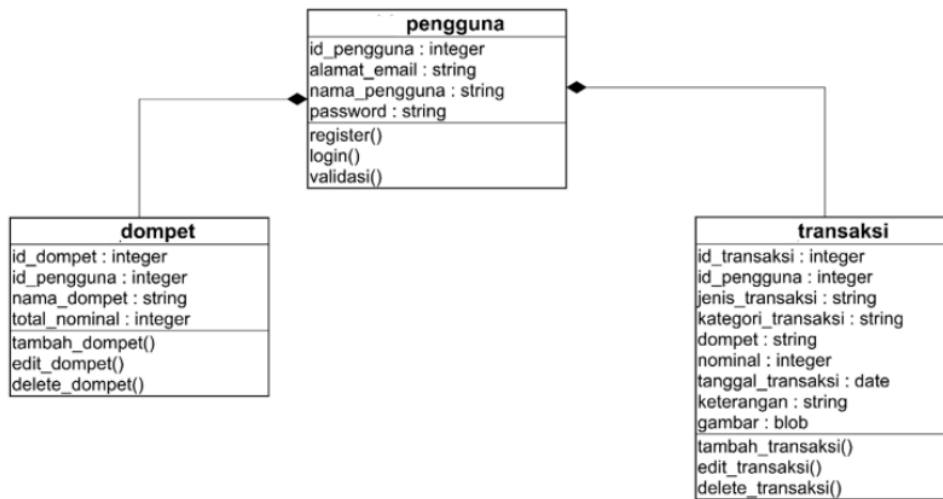
2 Untuk menggambarkan struktur sistem maka digunakan *class diagram* yang menggambarkan *class-class* di dalam sebuah sistem yang akan dibangun dan berhubungannya antara satu dengan yang lain, serta dimasukkan pula operasi, atribut, sistem dan method [6]. *Class diagram* menggunakan beberapa simbol, sebagai berikut:

Tabel 3 Simbol *Class Diagram* [23]

| No | Simbol | Nama | Keterangan |
|----|--------|-------------------------|---|
| 1. | ———— | <i>Association</i> | Menyatakan hubungan antar <i>class</i> , biasanya disertai juga dengan <i>multiplicity</i> . |
| 2. | ◇ | <i>Nary Association</i> | Simbol ini digunakan untuk menghindari asosiasi yang lebih dari dua objek. |
| 3. | -----> | <i>Dependency</i> | Hubungan di mana perubahan yang terjadi pada elemen <i>independent</i> akan mempengaruhi elemen yang bergantung padanya (elemen <i>dependent</i>). |
| 4. | ←----- | <i>Realization</i> | Operasi yang dilakukan oleh suatu objek. |

| | | | |
|----|---|-----------------------|--|
| 5. |  | <i>Class</i> | Kumpulan dari objek-objek yang berbagi atribut dan operasi yang sama. |
| 6. |  | <i>Generalization</i> | Menggambarkan hubungan generalisasi dan spesialisasi antara dua buah <i>class</i> dimana fungsi yang satu lebih umum kedudukannya dibandingkan dengan lainnya. |
| 7. |  | <i>Aggregation</i> | Menunjukkan hubungan antara <i>class</i> dengan posisi <i>whole part</i> . |

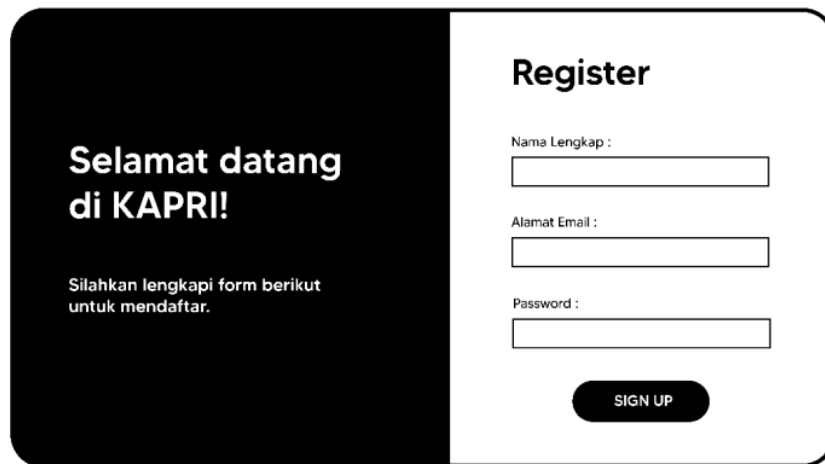
Berdasarkan struktur serta deskripsi dari *class* yang telah dikelompokkan, atribut setiap *class*, metode, dan hubungan dari setiap objek, maka *class diagram* dari sistem yang dibangun dapat digambarkan sebagai berikut:



Gambar 4. *Class diagram*

Pada Gambar 4 di atas menampilkan apa saja yang terdapat di dalam sistem. Seperti nama *class*, nama atribut di dalam *class*, tipe data dari setiap atribut, dan relasi antara *class*.

Setelah perancangan fungsional dilakukan, maka dilakukan rancangan antar muka dengan tujuan untuk memenuhi kebutuhan para pengguna sistem serta memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada programmer [22]. Berikut ini beberapa rancangan antar muka yang dapat memberikan gambaran fitur-fitur dalam aplikasi yang dibangun secara detail.



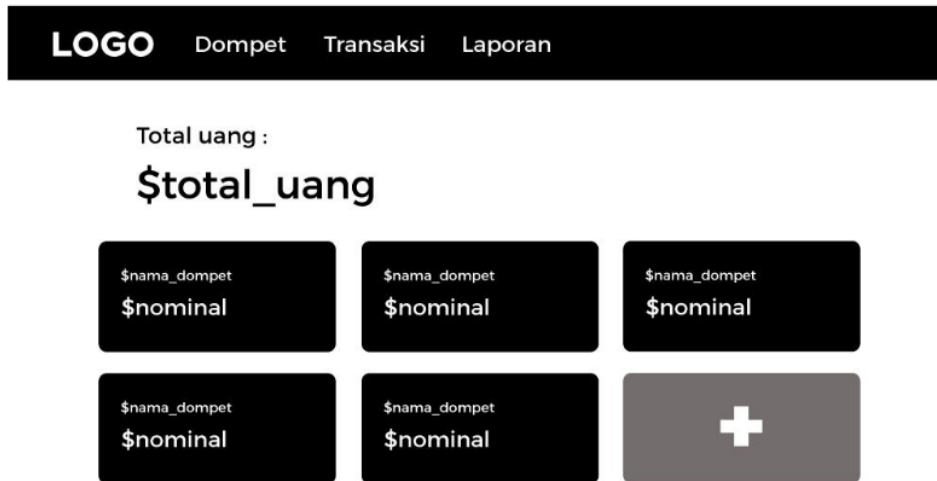
Gambar 5. Rancangan antar muka: *Register*

Gambar 5 di atas merupakan rancangan antar muka yang digunakan *user* untuk melakukan registrasi dengan memasukkan nama lengkap, alamat *email*, dan *password* yang akan digunakan untuk mengakses aplikasi.



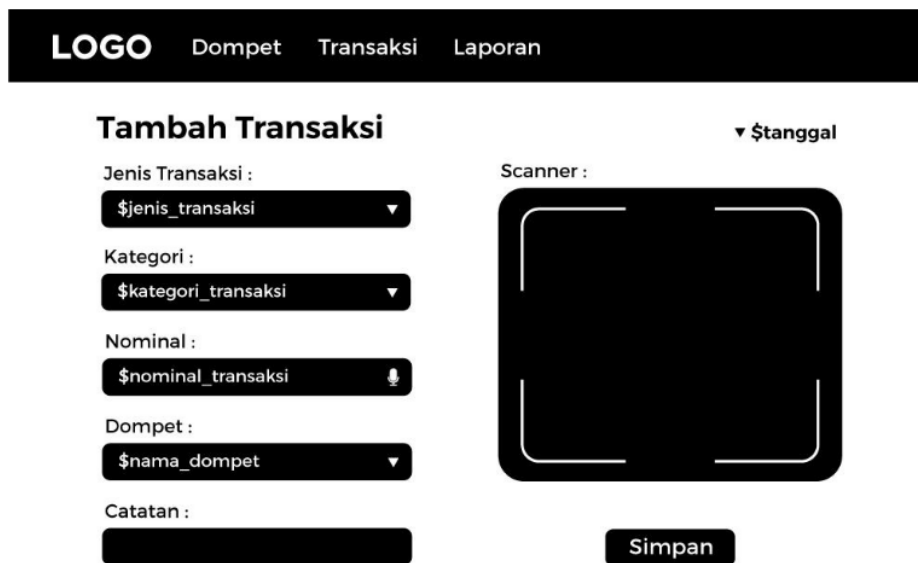
Gambar 6. Rancangan antar muka: *index*

Gambar 6 di atas adalah gambaran mengenai antar muka dari menu *home* yang ada pada sistem. Menampilkan total uang, pemasukan bulanan, pengeluaran bulanan, dompet yang digunakan, dan laporan keuangan selama 6 bulan terakhir.



Gambar 7. Rancangan antar muka : Dompets

Pada gambar 7 di atas terdapat gambar rancangan antar muka halaman dompets. Halaman ini akan menampilkan data-data dari setiap dompets yang telah ada di dalam *database*.



Gambar 8. Rancangan antar muka : scan

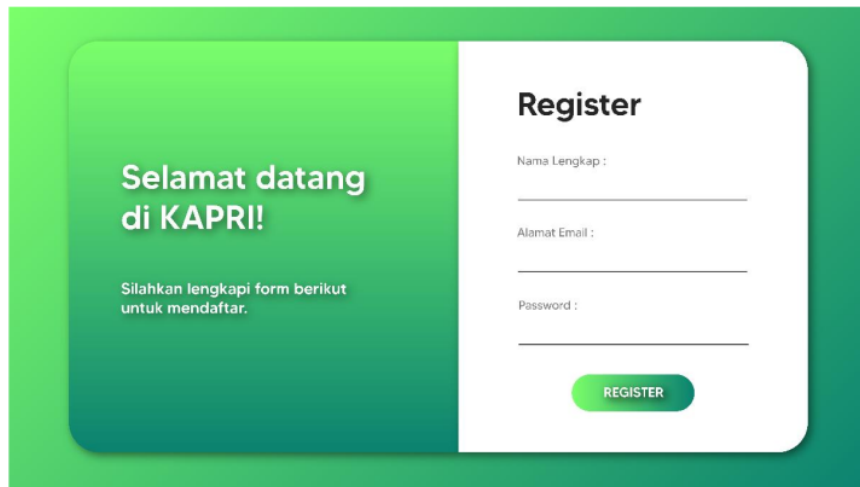
Pada Gambar 8 di atas, rancangan antar muka dari fitur *scan* yang diharapkan dapat memudahkan pengguna dalam memasukkan nominal uang. *Form* yang ada terdiri dari jenis transaksi, kategori, nominal, jenis dompets dan catatan.



Gambar 9. Rancangan antar muka : Laporan

Gambar 9 menampilkan rancangan antar muka laporan. Terdapat diagram untuk menampilkan laporan mingguan dari pengeluaran pengguna. Dalam laporan keuangan tertera keterangan pengeluaran, nominal yang dikeluarkan, kategori dan juga tanggal dari pembuatan pengeluaran tersebut.

Berdasarkan rancangan antar muka yang telah buat, selanjutnya akan dilakukan pemrograman sistem sesuai dengan yang telah dirancang sebelumnya. Berikut ini hasil implementasi dari rancangan yang telah dilakukan dalam pembuatan sistem.



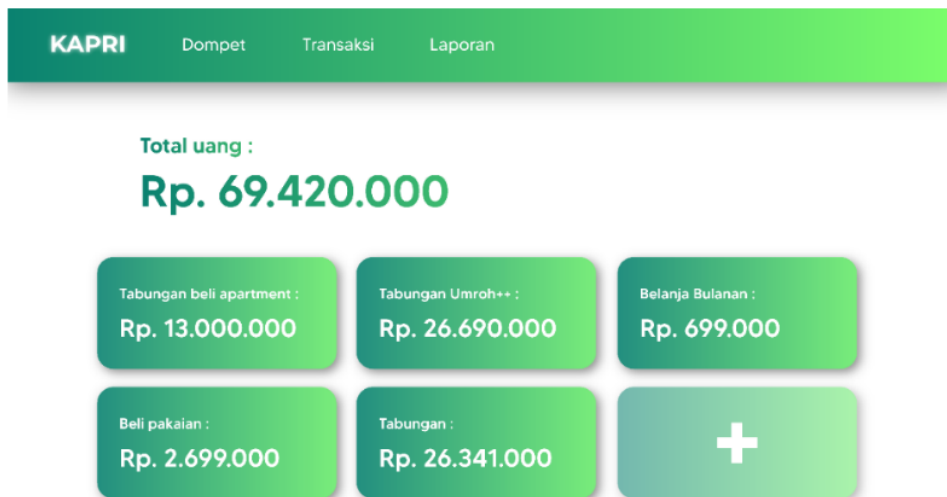
Gambar 10. Hasil implementasi: Register

Pada Gambar 10 di atas dapat terlihat bahwa setiap *user* yang akan menggunakan aplikasi pemanfaatan citra digital dalam manajemen keuangan pribadi berbasis *website* ini harus diregistrasi terlebih dahulu di dalam sistem.



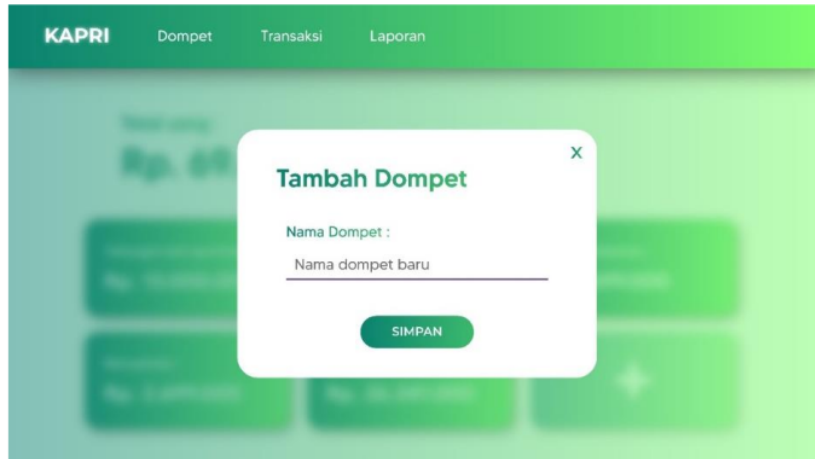
Gambar 11. Hasil Implementasi: *Index*

Gambar 11 di atas merupakan tampilan utama dari aplikasi ini. Berguna untuk menampilkan data secara singkat dari 3 halaman yang berbeda. Jika *user* memilih menu *dompot*, maka *user* akan dialihkan ke tampilan menu *dompot* seperti yang terlihat pada Gambar 12 berikut ini.



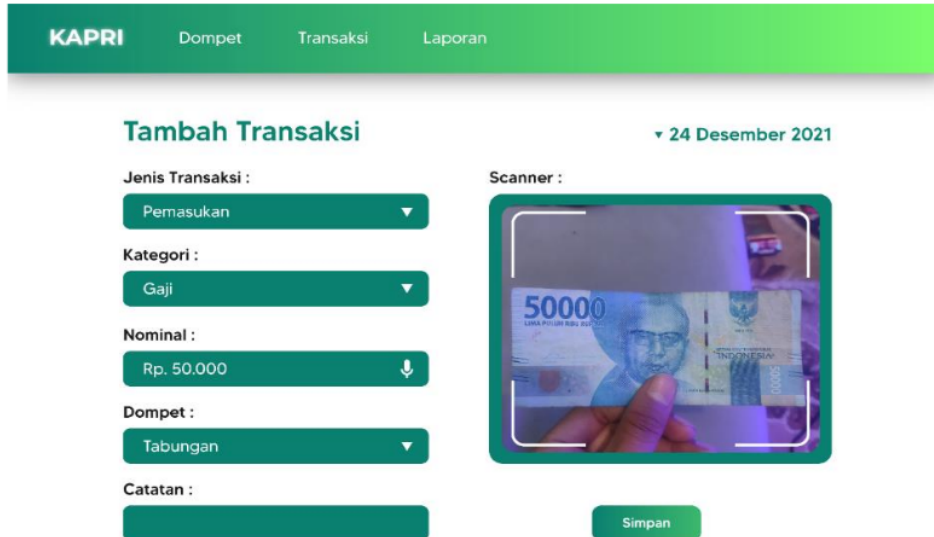
Gambar 12. Hasil Implementasi: *Dompot*

Gambar 12 merupakan hasil implementasi dari rancangan yang terdapat pada gambar 5, di mana data yang ditampilkan adalah data yang ditambahkan oleh *user* agar bisa dikelola oleh sistem. Untuk menambahkan data pada Gambar 12 harus melalui proses menambahkan dompet.



Gambar 13. Hasil Implementasi: Tambah Dompet

Pada gambar 13 menunjukkan mekanisme menambahkan dompet yang akan ditampilkan di menu dompet. Penampilan data tersebut bertujuan agar user dapat dengan mudah memantau keuangan pribadi ketika ada pengurangan karena adanya transaksi atau pemasukan keuangan. Sedangkan pada tampilan terakhir, digunakan *user* jika akan melakukan transaksi.



Gambar 14. Hasil Implementasi: Tambah Transaksi

Gambar 14 menunjukkan *Form* tambah transaksi yang ada terdiri dari jenis transaksi, pemilihan kategori dompet, menentukan nominal dan juga menentukan kategori transaksi. Pada *form* ini pula teknologi citra digital digunakan untuk men *scan* uang atau struk belanja, sehingga *input* data dapat dilakukan dengan cepat dan akurat. Setelah *user* menambahkan transaksi, data tersebut akan masuk ke menu laporan. Menu laporan berisi data laporan transaksi yang terdahulu agar *user* dapat melihat pengeluaran atau pemasukan di akun miliknya.

Untuk mengimplementasikan aplikasi pemanfaatan citra digital dalam manajemen keuangan pribadi berbasis *web* ini, dibutuhkan beberapa perangkat, yakni perangkat keras (*hardware*), perangkat lunak (*software*), dan sumberdaya manusia (*brainware*). Spesifikasi perangkat keras yang digunakan untuk mengembangkan sistem maupun menjalankannya adalah sebagai berikut:

- 1) CPU, setara dengan pentium 4 atau sekelasnya
- 2) RAM, 10 GB DDR3
- 3) Harddisk, 500 GB
- 4) VGA, setara AMD radeon R4 graphics
- 5) Monitor
- 6) *Keyboard*
- 7) *Mouse*

Kemudian, spesifikasi perangkat lunak yang digunakan untuk menjalankan sistem yang telah dibangun terdiri dari:

- 1) Sistem operasi: windows versi 8 atau 10
- 2) Sublime text untuk menulis *code* program
- 3) XAMPP, untuk menjalankan localhost dan PHPMyAdmin,
- 4) *Browser*, seperti google chrome, Mozilla Firefox, dan lain-lain

Sedangkan sumberdaya manusia yang dibutuhkan untuk menggunakan sistem ini adalah cukup dengan *user* yang sudah terbiasa mengoperasikan komputer dan internet.

Kesimpulan

Berdasarkan pemaparan tersebut di atas, pemanfaatan citra digital dalam pengelolaan keuangan pribadi berbentuk fitur *scan* uang dapat membantu dalam mengatur keuangan pribadi, seperti mengatur pemasukkan dan pengeluaran uang pribadi. Dengan adanya fitur tersebut pengguna akan lebih mudah dalam melakukan *input* data pemasukkan dan pengeluaran menggunakan fitur *scan* data melalui kamera dan *scan* data melalui gambar. Sistem ini mampu diakses melalui berbagai sistem operasi yang memiliki koneksi internet dan *browser*, sehingga memudahkan pengguna dalam memanfaatkan berbagai fasilitas yang ada dalam sistem. Sistem ini dapat dikembangkan lebih jauh dengan menambahkan fitur yang dapat mengingatkan pengguna dalam penggunaan uang, seperti notifikasi limit penggunaan keuangan.

Ucapan Terima Kasih

Terima kasih disampaikan kepada Lembaga Penelitian dan Pengabdian Pada Masyarakat (LPPM) Institut Digital Ekonomi LPKIA dalam dukungannya terhadap penelitian dan pembimbingan penulisan karya ilmiah.

Daftar Pustaka

- [1] L. Dewi *dkk.*, “Studi Komparasi Alat Bantu Pengelola Keuangan Pribadi Berbasis Mobile Comparative Study Mobile Based,” vol. 2, no. 1, 2019.
- [2] M. Setiawan, “Aplikasi Pencatatan Keuangan Pribadi Dengan Menggunakan Notifikasi Dan Infografis Berbasis Android,” Feb 2021.

- [3] G. C. Antonio, R. Intan, dan R. Adipranata, "Pengelolaan Keuangan Pribadi yang Interaktif Berbasis Android," *Jurnal Infra*, vol. 9, no. 2, hlm. 144–148, Okt 2021.
- [4] A. Susanto, A. Noertjahyana, dan A. Setiawan, "Aplikasi Pengelola Keuangan Pribadi Berbasis Android," *Jurnal Infra*, vol. 4, no. 2, hlm. 224–227, Jul 2016.
- [5] S. Ratna, "Aplikasi Pengelola Keuangan Pribadi Berbasis Desktop," *Technologia: Jurnal Ilmiah*, vol. 12, no. 2, hlm. 68, 2021, doi: 10.5/JS/jquery.datatables.min.js.
- [6] D. Uzairi, I. Najiyah, K. Kunci -Android, dan M. Keuangan, "Aplikasi Manajemen Keuangan Berbasis Android Dengan Fitur Reminder Dan Push Notification," *eProsiding Teknik Informatika (PROTEKTIF)*, vol. 1, no. 1, hlm. 195–199, Jul 2021.
- [7] Novryan dan A. Budi, "Implementasi Aplikasi Pencatatan Keuangan Pribadi Berbasis Android," *Jurnal Informatika dan Bisnis*, vol. 5, no. 2, 2016.
- [8] M. G. Priyantono, "Pembangunan Aplikasi Perencanaan Keuangan Pribadi Menggunakan Teknologi Firebase Cloud Messaging Dan Api Toko Online Berbasis Android," Agu 2019.
- [9] D. M. C. B. Ginting, "Rancang Bangun Aplikasi Pengelola Keuangan (Money Manager) di LKP Oxford Binjai Berbasis Android," *Kumpulan Karya Ilmiah Mahasiswa Fakultas sains dan Teknologi*, vol. 2, no. 2, hlm. 157–157, Jun 2020.
- [10] C. T. Lestari dan F. Latifah, "Aplikasi Pencatatan Keuangan Pribadi Dengan Analisa SWOT Menggunakan Algoritma Sequential Search Berbasis Mobile," *Journal of Information System, Applied, Management, Accounting and Research*, vol. 3, no. 2, hlm. 11–18, Mei 2019.
- [11] F. I. Nurzamziah, "Aplikasi Manajemen Keuangan Pribadi Berbasis Android," 2020. <http://repota.jti.polinema.ac.id/460/> (diakses Nov 14, 2021).
- [12] S. Ratna, "Aplikasi Pengelola Keuangan Pribadi Berbasis Desktop," *Technologia: Jurnal Ilmiah*, vol. 12, no. 2, hlm. 68–71, Apr 2021, doi: 10.31602/TJLV12I2.4572.
- [13] V. S. Febriyanny, "Rancang Bangun Pengatur Keuangan Pribadi Dengan Menggunakan Chatbot (Studi Kasus Pada Mahasiswa Unikom)," Agu 2019.
- [14] S. R. Sulistiyanti, F. X. A. Setyawan, dan M. Komarudin, *Pengolahan Citra, Dasar dan Contoh Penerapannya*. Teknosain, 2016.
- [15] RD. Kusumanto dan A. N. Tompunu, "Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB," *Semantik*, vol. 1, no. 1, Apr 2011.
- [16] S. Munawaroh dan F. A. Sutanto, "Pengolah Citra Digital untuk Identifikasi Uang Kertas," <https://www.unisbank.ac.id/ojs/index.php/jti1/article/view/109/104>, vol. 15, hlm. 34–40, 2010.
- [17] I. Mahendra, D. Tresno, dan E. Yanto, "Agile Development Methods Dalam Pengembangan Sistem Informasi Pengajuan Kredit Berbasis Web (Studi Kasus : Bank BRI Unit Kolonel Sugiono)," *Jurnal Teknologi Dan Open Source*, vol. 1, no. 2, hlm. 13–24, Des 2018.
- [18] N. Lutfiani, E. Purnama Harahap, Q. Aini, A. Dimas Ahsanul Rizki Ahmad, dan U. Rahardja, "Inovasi Manajemen Proyek I-Learning Menggunakan Metode Agile Scrum Cite this paper," *Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 5, no. 1, hlm. 96–101, Sep 2020, doi: 10.30743/infotekjar.v5i1.2848.
- [19] Diah Puspitasari, "Sistem Informasi Perpustakaan Sekolah Berbasis Web," *Skripsi*, vol. 12, no. 2, hlm. 8–35, Jul 2016.
- [20] M. Efendi dan M. Yudhati, "Pengaruh Ekspektasi Laba Terhadap Minat Berwirausaha (Studi Pada Mahasiswa Angkatan 2013 STIE MBI Jakarta)," *Jurnal Administrasi Dan Kebijakan Publik*, vol. 7, no. 2, hlm. 42–64, Agu 2017, doi: 10.33558/AKP.V7I2.606.
- [21] Y. Utama, "Sistem Informasi Berbasis Web Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Sriwijaya," *JSI: Jurnal Sistem Informasi (E-Journal)*, vol. 3, no. 2, hlm. 359–370, Okt 2011, doi: 10.36706/JSI.V3I2.738.
- [22] F. Ampu, "Perancangan Sistem Informasi Reservasi Kamar Berbasis Web Pada Hotel Karmila Bandung," *Jurnal Sistem Informatika dan Informasi*, vol. 1, hlm. 1–476, 2019.

- [23] Lily Puspa Dewi, Uce Indahyanti, dan Yulius Hari S, “Pemodelan Proses Bisnis Menggunakan Activity Diagram UML Dan BPMN (Studi Kasus FRS Online) ,” 2012.
- [24] L. Hertati,” The Role of Accounting Information Budgets on Cost Responsibility Centers”,
Jurnal *@is The Best: Accounting Information Systems and Information Technology Business Enterprise*,
vol. 6 no.2, hlm. 121-135, 2021

Pengelolaan Keuangan Pribadi Menggunakan Teknologi Citra Digital Berbasis Website.pdf

ORIGINALITY REPORT

14%

SIMILARITY INDEX

14%

INTERNET SOURCES

7%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

ojs.unikom.ac.id

Internet Source

11%

2

ojs.stmik-banjarbaru.ac.id

Internet Source

3%

Exclude quotes Off

Exclude matches < 2%

Exclude bibliography On