

Implementasi *Progressive Web Apps (PWA)* pada *Repository E-Portofolio* Mahasiswa

Samsul Aripin¹, Somantri²

Teknik Informatika
Universitas Nusa Putra
Sukabumi, Indonesia

e-mail: ¹samsul.aripin_ti17@nusaputra.ac.id, ²somantri@nusaputra.ac.id

Diajukan: 14 Januari 2021; Direvisi: 25 Maret 2021; Diterima: 31 Maret 2021

Abstrak

Setiap mahasiswa memiliki berbagai informasi terkait dengan kegiatan tri dharma yang pernah dilakukannya, seperti prestasi akademik, aktivitas penelitian, publikasi karya ilmiah, pengabdian kepada masyarakat, prestasi, penghargaan, kegiatan organisasi, sertifikasi dan aktivitas pendukung lainnya. Sering kali data-data terkait aktivitas-aktivitas tersebut sulit ditemukan ketika dibutuhkan, maka dari itu dibutuhkan sebuah repository yang dapat menghimpun informasi tersebut. Melalui repository portofolio online setiap mahasiswa dapat membuat E-Portofolio yang mengompilasikan seluruh aktivitas tri dharma yang pernah dilakukannya selama masa perkuliahan, E-Portofolio tersebut dapat dipamerkan ke berbagai pihak sehingga akan memotivasi mahasiswa untuk terus mengikuti berbagai kegiatan yang dapat dipajang di portofolionya, portofolio tersebut juga dapat digunakan untuk berbagai kebutuhan seperti pengajuan beasiswa, persyaratan magang, dan lain-lain. Pengembangan perangkat lunak yang dilakukan pada penelitian ini menggunakan metode Waterfall, Sedangkan perancangan sistem yang digunakan adalah berbasis Progressive Web Apps (PWA) yang merupakan teknologi website modern yang membuat website dapat dijalankan seperti aplikasi native. Hasil pengujian menunjukkan sistem sudah berjalan normal, mudah digunakan, dan memenuhi kriteria progressive web apps seperti dapat dijalankan secara offline dan dapat diakses melalui home screen.

Kata kunci: Repository, E-Portofolio, Progressive Web Apps, Mahasiswa, Prestasi.

Abstract

Each student has various information related to tri dharma activities that have been carried out, such as academic achievements, research activities, publication of scientific papers, community service, achievements, awards, organizational activities, certification and other supporting activities. Often data related to these activities is difficult to find when needed, therefore a repository is needed that can collect this information. Through the online portfolio repository each student can create an E-portfolio that compiles all the tri dharma activities he has done during the lecture period, the E-portfolio can be exhibited to various parties so that it will motivate students to continue to participate in various activities that can be displayed in their portfolios, the portfolio is also can be used for various needs such as applying for scholarships, internship requirements, and others. The software development carried out in this study uses the Waterfall method, while the system design used is based on Progressive Web Apps (PWA) which is a modern website technology that makes websites run like native applications. The test results show that the system is running normally, is easy to use, and meets the criteria set by the progressive web application that can be run offline and can be accessed via the home screen.

Keywords: Repository, E-Portfolio, Progressive Web Apps, Student, Achievement.

1. Pendahuluan

Portofolio *online* atau E-Portofolio sebagai wadah untuk memamerkan berbagai hasil prestasi seseorang telah banyak digunakan. Beberapa contoh portofolio hasil penelitian (Academia.edu dan LinkedIn.com), Karya seni (deviantart.com, behance.com, kreabi.com), dan lain-lain. dengan E-Portofolio data prestasi dapat disimpan dengan aman dan dapat dipublikasikan [1].

Sebagai salah satu institusi pendidikan perguruan tinggi membutuhkan sebuah sistem yang mengelola E-Portofolio mahasiswanya [2], hal ini menjadi kebutuhan khusus karena portofolio yang

didokumentasikan oleh perguruan tinggi akan berbeda dengan portofolio yang dipajang secara individu di sosial media sebagai seorang profesional/pekerja, maka E-Portofolio mahasiswa dibuat untuk mendokumentasikan seluruh prestasi mahasiswa selama masa perkuliahan baik prestasi yang didapat dari internal perguruan tinggi maupun dari luar perguruan tinggi. adapun contoh dari data yang akan didokumentasikan pada E- portofolio mahasiswa adalah: data pribadi, data akademik, karya ilmiah, penelitian, pengabdian kepada masyarakat, publikasi jurnal, publikasi oral/poster, paten dan HAKI bahan ajar/buku/*software*, pengalaman jabatan, pengalaman magang, praktek kerja lapangan, penghargaan, pengalaman organisasi, kepanitiaan, dan lain-lain.

E-Portofolio sendiri akan lebih efektif jika dikelola secara mandiri oleh masing-masing mahasiswa, Manfaat E-Portofolio yang dikelola secara mandiri oleh mahasiswa diharapkan dapat merangsang minat belajar dan keterampilan mahasiswa, dan dapat merekam setiap pencapaian mahasiswa dengan baik [3]. Manfaat lain dari E-Portofolio mahasiswa sendiri adalah sebagai sarana dokumentasi kegiatan yang bersifat keaktifan yang dilakukan selama perkuliahan, hal ini dibutuhkan karena seorang mahasiswa biasanya sering melupakan bukti prestasi apa saja yang didapat selama perkuliahan dan tidak mengarsipkannya dengan baik, sehingga sulit jika dibutuhkan di kemudian hari. Dengan adanya E-Portofolio mahasiswa, mahasiswa dapat mengunggah seluruh prestasinya dan dapat diambil sewaktu-waktu jika dibutuhkan, seperti untuk pengajuan beasiswa dan pengajuan magang.

Berdasarkan permasalahan di atas maka diusulkan untuk membuat sebuah *repository* E-Portofolio yang akan mendokumentasikan seluruh prestasi dan kegiatan keaktifan mahasiswa selama perkuliahan di sebuah perguruan tinggi. Adapun *framework* yang akan digunakan dalam membangun *website* tersebut adalah Laravel, dan dibangun dengan konsep *progressive web apps* (PWA). PWA memiliki keunggulan dibandingkan aplikasi *website* seperti dapat diakses dengan lebih cepat, mendukung fitur *offline*, dan dapat berjalan layaknya aplikasi *native* [4]. Teknologi PWA ini cocok digunakan pada *repository* E-Portofolio Mahasiswa, karena dengan satu kali proses pembuatan dapat menghasilkan aplikasi *web*, *desktop*, dan Android secara bersamaan.

2. Kajian Teori

2.1. E-Portofolio

Istilah portofolio berasal dari kata *port* yang berarti laporan dan *folio* yang berarti penuh atau lengkap [1]. dengan perkembangan teknologi yang pesat berkembanglah portofolio elektronik, yang lebih dikenal dengan istilah E-Portofolio. E-Portofolio merupakan halaman portofolio yang disusun secara *online* oleh pengguna, biasanya berbasis *website*, atau Android. *Input* dari E-Portofolio tersebut dapat mencakup teks *file* elektronik, Gambar, multimedia, *blog entries*, video animasi, dan poster. jika dipandang dari sudut pada milik mahasiswa, portofolio diri adalah laporan mengenai kualifikasi mahasiswa. Pembuatan portofolio mahasiswa bertujuan untuk mengumpulkan informasi produk atau hasil karya mahasiswa tersebut di berbagai bidang [2]. Dengan adanya portofolio seorang mahasiswa dapat dengan mudah mempublikasikan kualifikasinya kepada masyarakat dan dapat menggunakan portofolio tersebut jika sewaktu-waktu dibutuhkan.

2.2. Repository

Dalam istilah kepastakawanan, *repository* adalah tempat di mana data atau informasi disimpan, dipelihara, dan digunakan [5]. Dengan menggunakan *repository* data dapat dikelola dengan rapi sehingga akan mudah ditemukan jika sewaktu-waktu dibutuhkan, jika dahulu *repository* berbentuk arsip kumpulan kertas, keping CD, dan *Hardisk*, maka sekarang data- data tersebut hadir dalam bentuk *repository* digital. Institusi seperti perguruan tinggi juga membutuhkan *repository* E-Portofolio untuk mengarsipkan data-data mahasiswanya, dengan adanya *repository* E-Portofolio akan bermanfaat untuk penyebarluasan karya ilmiah mahasiswanya sehingga perguruan tinggi dapat meningkatkan kuantitas hasil penelitian dan berdampak pada peningkatan kredibilitas dan akreditasi suatu perguruan tinggi [6]. Manfaat lain dari penggunaan *repository* institusi adalah mengurangi biaya dan ruang yang biasa digunakan dalam *repository* berbentuk arsip kertas, data juga dapat disimpan dengan aman karena disimpan secara *online* sehingga dapat terhindar dari kerusakan fisik seperti kebakaran.

2.3. Framework Laravel

Laravel adalah *framework* pembuatan *website* yang dibuat oleh Taylor Otwell yang dibangun dengan konsep MVC (*Model- View-Controller*), dan dilengkapi juga *command line tool* yang bernama “*Artisan*” yang bisa digunakan untuk *packaging bundle* dan instalasi *bundle* melalui *command prompt* [7]. Laravel dibangun menggunakan bahasa pemrograman PHP yang cukup populer di Indonesia. Selain itu Jika dibandingkan dengan *framework* PHP lainnya Laravel memiliki banyak keunggulan seperti: *auth*

scaffolding yang memudahkan pembuatan fitur *login* dan *register* secara otomatis tanpa harus menulis kode program, dan *Eloquent ORM* untuk mengelola *database* tanpa menggunakan bahasa SQL, dan cocok untuk membuat layanan *restfull API* dan *Repository online* [8].

2.4. Progressive Web Apps (PWA)

Progressive Web Apps (PWA) adalah konsep penerapan pembuatan *website* yang menerapkan teknologi *service worker*, *web manifest*, dan *cache API* [8]. Dengan membangun aplikasi menggunakan konsep PWA akan membuat aplikasi web berjalan di berbagai *platform* seperti *website*, *desktop*, dan *platform mobile* atau Android, PWA sendiri memiliki berbagai keunggulan seperti:

1. PWA memiliki sifat aplikasi *native* layaknya Android, tetapi tidak memerlukan spesifikasi *hardware* yang tinggi dalam membuatnya.
2. PWA dapat mempercepat proses pembuatan *website* dan mengurangi beban *server*.
3. PWA dapat berfungsi secara *offline*, Tidak seperti *website* yang tidak dapat diakses jika tidak ada koneksi internet.
4. PWA sudah banyak digunakan di perusahaan besar seperti Twitter, Facebook, Bukalapak, Tokopedia, dan sebagainya.

Proses instalasi PWA sangat mudah, hanya dengan mengakses *website* PWA yang diinginkan kemudian memilih opsi *install*, maka *icon* aplikasi PWA akan muncul di *home screen* [4]. Sebuah aplikasi PWA harus dibuat seringan mungkin agar dapat menghasilkan pengalaman pengguna dengan baik, agar *website* tidak melakukan proses *loading* terlalu lama. Berbagai cara dapat dilakukan dalam meningkatkan performa PWA di antaranya dengan memperkecil ukuran gambar yang digunakan pada *website*.

2.5. Service Worker

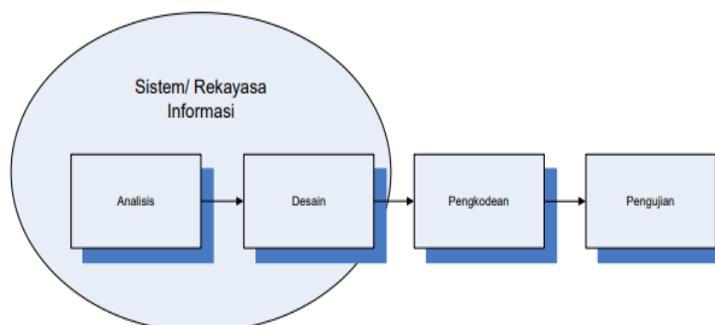
Pada umumnya sebuah *website* tidak dapat ditampilkan secara *offline* namun tidak demikian jika *website* menerapkan *service worker* pada *website*. *Service worker* adalah kode Javascript yang dijalankan oleh *browser* di latar belakang, dan terpisah dengan skrip lainnya. *Service worker* memiliki kemampuan untuk menyimpan aset-aset sebuah *website* seperti *file* HTML, CSS, Javascript, dan gambar di penyimpanan *browser*, sehingga ketika *website* tersebut diakses tanpa koneksi internet *service worker* akan mengembalikan aset *website* yang sudah disimpan sebelumnya, hal ini membuat sebuah *website* dapat bekerja layaknya aplikasi *native* Android dan *desktop* [9]. Dengan kemampuan menyimpan *asset website* secara lokal, *service worker* juga dapat meningkatkan kecepatan akses sebuah *website* dan menghemat data.

2.6. Push Notification

Push notification adalah penerapan teknologi komunikasi yang memungkinkan aplikasi *server* untuk menyampaikan pesan singkat kepada pengguna. *push notification* dikirimkan ke *desktop* atau layar telepon pengguna kapan pun ketika web *browser* dijalankan meskipun pengguna membuka halaman web [10]. *Push notification* sering kali diterapkan ke dalam *progressive web apps* hal ini dilakukan karena penerapan *push notification* membutuhkan *service worker* yang juga diterapkan pada *progressive web apps*. Penerapan *push notification* membutuhkan *server* penyedia layanan web push, salah satu penyedia layanan *push notification* yang sering digunakan adalah Firebase Cloud Messaging (FCM).

3. Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini adalah metode air terjun atau *waterfall*. Metode *waterfall* adalah metode yang tersusun (sekuensial) di mana *input* dari proses selanjutnya merupakan *output* dari proses sebelumnya [11]. Tahapan metode *waterfall* dapat dilihat pada Gambar 1.



Gambar 1. Metode *waterfall* [12].

Metode *waterfall* sangat cocok digunakan dalam pembangunan suatu aplikasi karena memiliki tahapan yang tersusun. Metode *waterfall* memiliki tahapan-tahapan sebagai berikut [12]:

1. Analisis Kebutuhan

Pada tahap ini dilakukan pengumpulan data-data dan kebutuhan terkait penelitian. Pengumpulan data dilakukan untuk mengetahui fitur-fitur yang dibutuhkan pada aplikasi E-Portofolio dan teknologi apa yang cocok untuk membangun aplikasi tersebut. Pengumpulan data dapat dilakukan dengan melakukan wawancara dan kuesioner pada mahasiswa dan masyarakat, sedangkan pemilihan teknologi dilakukan dengan melakukan studi literatur dan membaca dokumentasi teknologi.

2. Desain

Desain sistem dilakukan agar pembangunan sistem dapat dilakukan secara terstruktur. Adapun desain yang dibuat meliputi arsitektur sistem, *use case diagram*, dan *activity diagram*. desain sistem dibuat menggunakan *website plantuml.com*.

3. Pengodean

Pada tahap ini, *software* yang sudah dirancang mulai dibuat menggunakan bahasa PHP sebagai *backend*, juga HTML dan CSS untuk merancang tampilan *front end*. Proses pengodean dimulai dengan instalasi *framework* Laravel, pembuatan *database* dan tabel melalui Laravel *migrations*, pembuatan fitur CRUD (*Create, Read, Update, Delete*), dan pembuatan tampilan *front end*.

4. Testing sistem

Pada tahap ini program digabung dan diuji secara lengkap untuk memastikan apakah sesuai dengan kebutuhan perangkat lunak, dan meningkatkan layanan sistem sebagai agar menjadi lebih baik lagi. *Testing* yang pertama kali dilakukan adalah *black box testing* untuk menguji proses kerja aplikasi, *black box testing* dilakukan ketika suatu fitur aplikasi selesai dibuat. *Testing* selanjutnya dilakukan menggunakan *lighthouse* untuk menguji fitur *progressive web apps* dan performa aplikasi dari sisi *client*, *lighthouse* testing hanya bisa dilakukan jika aplikasi sudah dapat diakses melalui *browser*. *Testing* terakhir yang dilakukan adalah *usability testing*, testing ini dilakukan untuk menguji kemudahan dalam menggunakan aplikasi, *usability testing* dilakukan jika proses pembangunan aplikasi sudah selesai dikerjakan.

4. Hasil dan Pembahasan

4.1. Analisis Kebutuhan Sistem

Pembuatan Aplikasi *repository* E-Portofolio Mahasiswa diawali dengan mengumpulkan informasi mengenai kebutuhan sistem, yang didapatkan dengan, observasi dan wawancara pada dosen dan mahasiswa Prodi Teknik Informatika. Berdasarkan hasil wawancara, didapatkan deskripsi kebutuhan sistem sebagai berikut:

1. *Repository* diharapkan dapat menyimpan data portofolio mahasiswa.
2. E-Portofolio dapat diisi secara mandiri oleh mahasiswa.
3. E-Portofolio dapat dilihat oleh publik.
4. *Repository* diharapkan dapat berjalan dalam bentuk *progressive web apps*.

4.2. Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna *repository* E-Portofolio pada sistem adalah sebagai berikut:

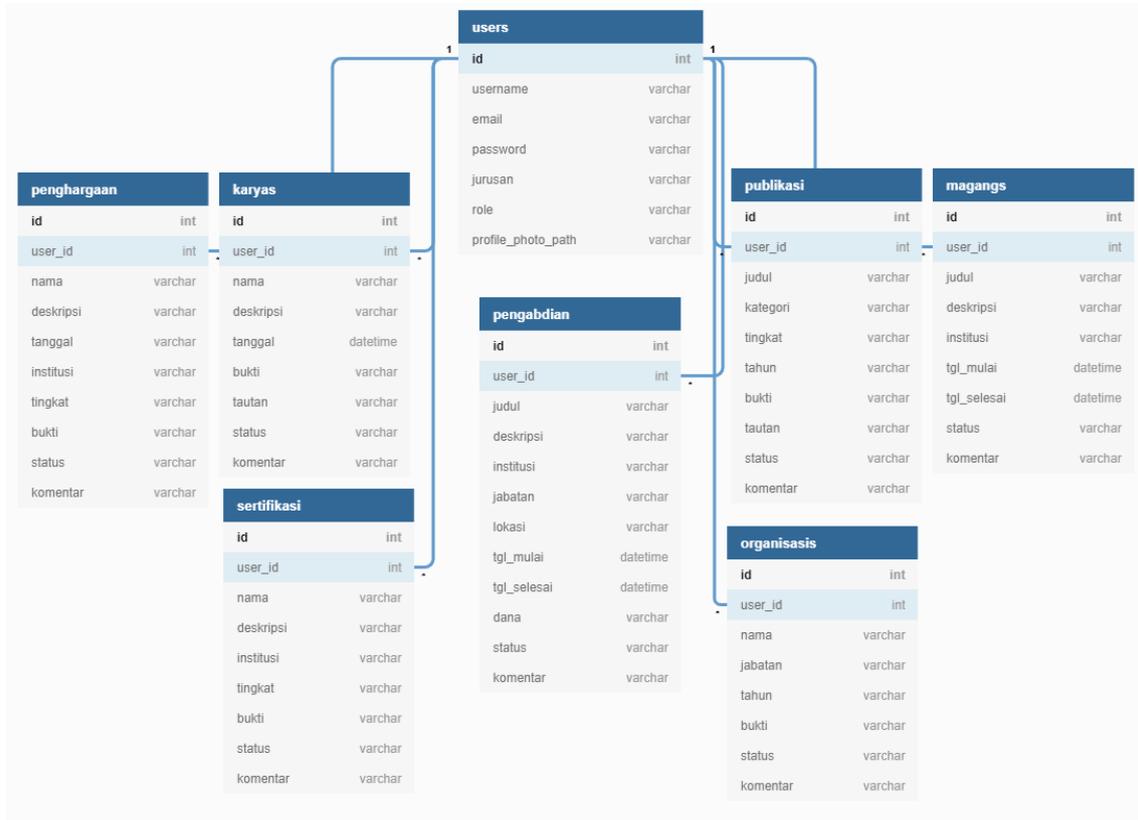
1. Mahasiswa dapat mengajukan data portofolio ke admin.
2. Mahasiswa dapat melihat, menambah, menghapus data portofolio miliknya.
3. Admin dapat menyetujui atau menolak data portofolio yang diajukan oleh mahasiswa.
4. Admin dapat melihat semua portofolio mahasiswa.
5. *Guest User* hanya dapat melihat portofolio mahasiswa yang sudah disetujui oleh admin.

4.3. Perancangan Sistem

Setelah dilakukan analisis kebutuhan pengguna maka proses selanjutnya adalah membuat perancangan sistem, perancangan sistem sendiri terdiri dari perancangan arsitektur sistem, *use case diagram*, dan *activity diagram*.

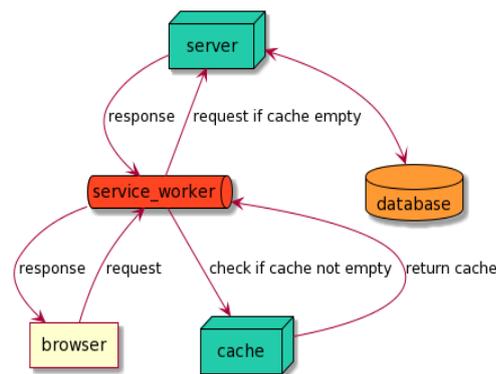
1. Desain Arsitektur Sistem

Desain Arsitektur sistem menjelaskan bagaimana setiap komponen dalam aplikasi saling berinteraksi. Sebelum membuat desain arsitektur sistem, diperlukan desain *relational database*. Database yang digunakan pada *repository E-Portofolio* adalah MySQL adapun tabel yang terdapat pada database merepresentasikan data-data portofolio, desain *relational database* dapat dilihat pada Gambar 2.



Gambar 2. Desain *relational database*.

Setelah membuat desain *relational database* selanjutnya dibuat desain arsitektur sistem, desain arsitektur dapat dilihat pada Gambar 3.



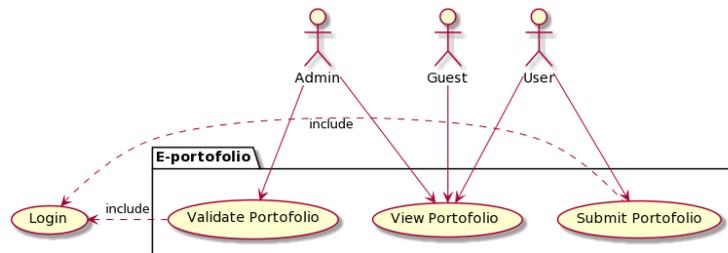
Gambar 3. Desain arsitektur sistem.

Pada arsitektur di atas menjelaskan bahwa pengguna mengakses aplikasi melalui *browser* lalu *browser* akan melakukan *request* ke *service worker*, setelah itu *service worker* akan memeriksa apakah terdapat konten yang tersimpan di *cache*, jika terdapat konten di *cache* maka konten tersebut akan dikembalikan ke *service worker* untuk ditampilkan di *browser*. Jika *service worker* tidak menemukan

konten yang di minta di dalam *cache* maka *service worker* akan melakukan *request* ke *server* kemudian *server* akan mengembalikan konten ke *service worker* untuk ditampilkan di *browser* dan disimpan di *cache*.

2. *Use case diagram*

Use case digunakan untuk memetakan fungsionalitas dari sistem sesuai dengan hasil analisis kebutuhan, juga memperlihatkan interaksi antara aktor yang menggunakan aplikasi serta aktivitas apa saja yang dapat aktor tersebut lakukan, serta memperlihatkan pekerjaan apa saja yang dapat dilakukan oleh beberapa jenis aktor yang berbeda. *Use case diagram* dapat dilihat pada Gambar 4.

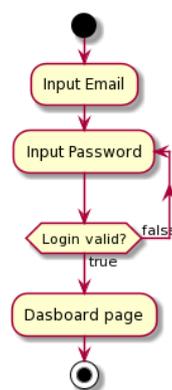


Gambar 4. *Use case diagram* Repository E-Portofolio Mahasiswa.

Pada diagram *use case repository* E-Portofolio mahasiswa diperlihatkan tiga orang aktor yaitu mahasiswa, admin, dan *guest*. Mahasiswa adalah *user* yang dapat *input* data portofolio, dan admin adalah aktor yang bertugas memvalidasi data portofolio yang di-*input*-kan oleh mahasiswa, jika data yang di-*input*-kan dianggap tidak valid maka admin akan mengirimkan pesan penolakan kepada mahasiswa, sedangkan jika data dianggap valid maka data portofolio dapat langsung dipublikasikan, sementara itu *guest* adalah aktor yang hanya dapat melihat portofolio yang telah dipublikasikan.

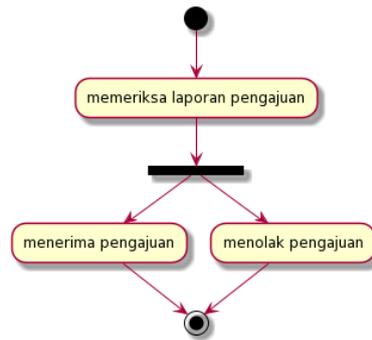
3. *Activity diagram*

Setelah membuat *use case diagram* selanjutnya dibuat *activity diagram*. *Activity diagram* digunakan untuk memetakan aktivitas yang terjadi pada setiap fungsionalitas yang tergambar pada *diagram use case*, aktivitas pertama yang dilakukan adalah *login*. Pada aktivitas *login* pengguna memasukkan *username* dan *password*, selanjutnya dilakukan validasi oleh sistem terhadap *username* dan *password* tersebut. Pengguna dapat menggunakan sistem jika *username* dan *password* dinyatakan valid, sebaliknya jika *username* atau *password* dianggap tidak valid maka pengguna akan diminta mengulangi proses *login*. Pada sistem ini tidak terdapat proses registrasi karena data akun pengguna sudah disediakan berdasarkan data mahasiswa dan admin Proses *login* dapat dilihat pada Gambar 5.



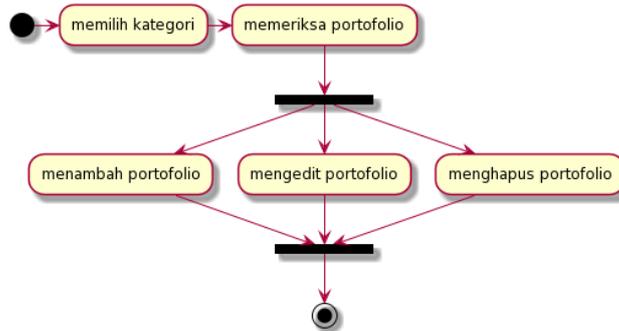
Gambar 5. *Activity diagram* Login.

Aktivitas selanjutnya adalah mengelola data portofolio oleh mahasiswa, mahasiswa dapat menambah (mengajukan data portofolio), mengedit, dan menghapus data portofolio, berikut tampilan, *activity diagram* mengelola portofolio ditunjukkan oleh Gambar 6.



Gambar 6. Activity diagram management portofolio.

Aktivitas selanjutnya adalah manajemen portofolio oleh admin, pada bagian ini admin akan memeriksa apakah ada pengajuan portofolio oleh mahasiswa, selanjutnya admin akan memvalidasi data ajuan apakah data tersebut dianggap sah atau tidak, jika data dianggap sah admin dapat mengirimkan pesan kepada pengirim ajuan bahwa data diterima dan data tersebut otomatis akan dipublikasikan ke dalam portofolio, sebaliknya jika data dianggap tidak sah maka admin akan mengirim pesan penolakan. Tampilan *management* portofolio oleh admin ditunjukkan oleh Gambar 7.



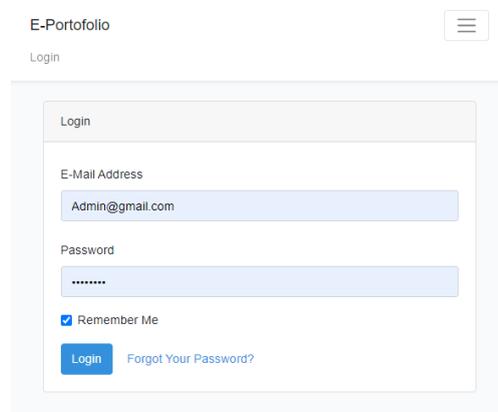
Gambar 7. Activity diagram manajemen portofolio oleh admin.

4.4. Pengodean Aplikasi

Jika rancangan sistem sudah dihasilkan maka proses selanjutnya adalah melakukan pengodean aplikasi, berikut hasil dari masing-masing pengodean pada pembuatan tampilan yang dilakukan.

1. Halaman *Login*

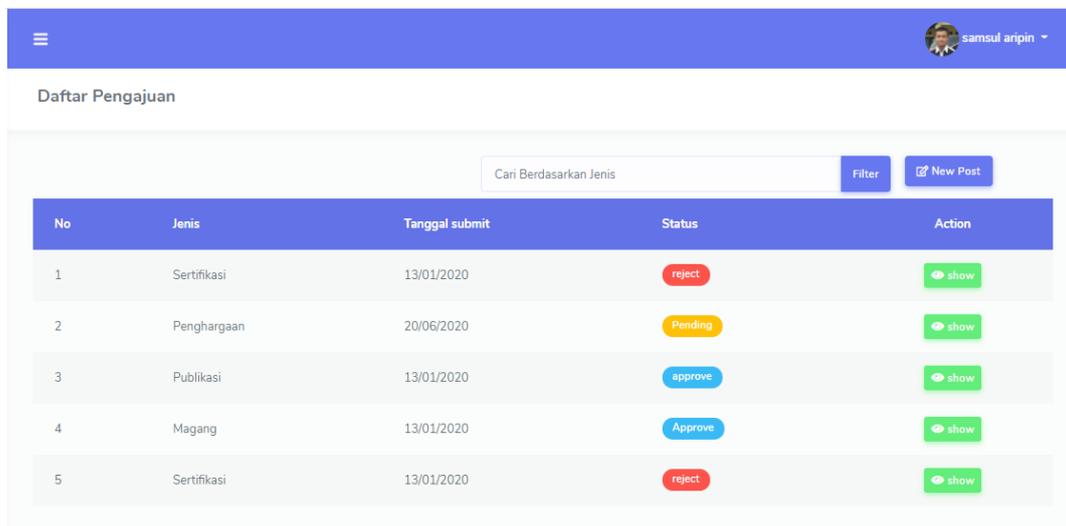
Untuk dapat melakukan pengolahan data portofolio baik mahasiswa maupun admin harus melakukan proses *login* terlebih dahulu, tampilan halaman *login* ditunjukkan pada Gambar 8.



Gambar 8. Halaman *Login* di web dan *smartphone*.

2. Halaman *Home*

Setelah proses *login* berhasil dilakukan maka pengguna akan diarahkan ke halaman *dashboard*, pada halaman ini mahasiswa akan melihat portofolionya yang berisi data diri, foto profil, serta, data portofolio yang terakhir kali di-*input*-kan, serta laporan pengajuan portofolio yang diterima, ditolak, dan masih dalam status menunggu. Tampilan halaman *dashboard* dapat dilihat pada Gambar 9.

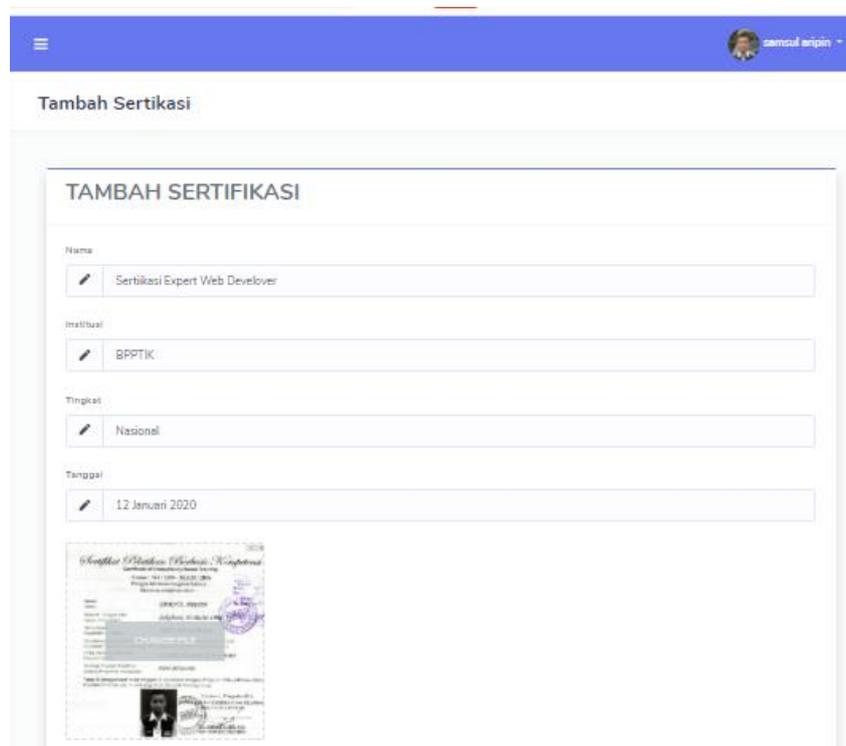


No	Jenis	Tanggal submit	Status	Action
1	Sertifikasi	13/01/2020	reject	show
2	Penghargaan	20/06/2020	Pending	show
3	Publikasi	13/01/2020	approve	show
4	Magang	13/01/2020	Approve	show
5	Sertifikasi	13/01/2020	reject	show

Gambar 9. Tampilan halaman Home di *website*.

3. Halaman *Input* Portofolio

Pada halaman ini diperlihatkan *form input* pengajuan portofolio baru oleh mahasiswa, *form input* data portofolio memiliki *input field* yang berbeda-beda tergantung dari kategori data portofolio yang akan diajukan, mahasiswa juga wajib melampirkan bukti untuk setiap data yang diajukan, berikut ini adalah contoh *form* pengajuan data sertifikasi ditunjukkan oleh Gambar 10.



TAMBAH SERTIFIKASI

Nama: Sertifikasi Expert Web Developer

Instansi: BPPTIK

Tingkat: Nasional

Tanggal: 12 Januari 2020

(Preview of a certification document is shown below the form fields.)

Gambar 10. Halaman *input* portofolio di *website*.

4. Halaman Daftar Portofolio

Pada halaman ini diperlihatkan tabel data pengajuan portofolio yang dikirimkan ke admin, mahasiswa dapat melihat status dari masing-masing ajuan, seperti terlihat pada Gambar 11.

No	Jenis	Pemohon	Tanggal submit	Status	Action
1	Sertifikasi	Samsul Aripin	13/01/2020	reject	Aksi
2	Penghargaan	Rizki Gunawan	20/06/2020	Pending	Aksi
3	Publikasi	Jamaludin	13/01/2020	approve	Aksi
4	Magang	Rudi Kurniawan	13/01/2020	Approve	Aksi
5	Sertifikasi	Ahmad Ridwan	13/01/2020	reject	Aksi

Gambar 11. Halaman Daftar Portofolio di Website.

Terdapat tiga status data ajuan yaitu diterima, ditolak, dan menunggu, mahasiswa juga dapat melihat pesan *review* dari admin untuk setiap pengajuan data.

5. Halaman *Panel Admin*

Pada halaman ini admin akan melihat daftar pengajuan data portofolio yang dikirim oleh mahasiswa, selanjutnya admin akan melakukan validasi terhadap data ajuan untuk selanjutnya data tersebut dapat diterima atau ditolak, Gambar 12 menunjukkan halaman panel admin.

Welcome Back, Samsul!
Selamat datang di Repository E-portofolio Mahasiswa input data kualifikasi anda sebanyak-banyaknya untuk memperkaya portofolio anda

	Submitted 187	Approve 30	reject 21
--	------------------	---------------	--------------

NIM: 16172008
Nama: Sudirman
Jenis Kelamin: Laki-laki
Jurusan: Teknik Informatika

Your recent post [View All >](#)

- penghargaan: Juara Debat Bahasa Inggris Nasional
- Sertifikasi: Google Develover Associated
- Pengabdian: Satgas Covid-19 Kab.sukabumi
- Organisasi: Sekertaris BNN Sukabumi
- Penelitian: Penilitia Dampak Pandemi Pada Minat Belanja Online

Gambar 12. Halaman panel admin *website*.

4.5. Testing

Setelah dilakukan pengodean aplikasi maka proses selanjutnya adalah melakukan testing pada aplikasi yang telah dibangun berikut adalah hasil testing yang telah dilakukan.

1. Black Box Testing

Black box testing dilakukan untuk menguji fungsional aplikasi, keunggulan *black box testing* adalah proses pengujian yang cepat karena testing hanya dilakukan pada tampilan luar dan fungsional aplikasi saja tanpa memperhatikan bagaimana proses *internal* bekerja. Dari hasil *black box testing*, hasil dari *black box testing* ditunjukkan pada tabel 1.

Tabel 1. Hasil *black box testing*.

Proses	Kasus Pengujian	Harapan	Hasil	Kesimpulan
Mahasiswa dan admin melakukan <i>login</i>	Masukan email dan <i>password</i> kemudian klik 'tombol' <i>login</i>	Masuk sistem jika <i>username</i> dan <i>password</i> benar, sistem menampilkan halaman sesuai dengan <i>role</i> pengguna	Sesuai	Normal
Mahasiswa memilih kategori portofolio	Melakukan klik pada daftar menu di <i>sidebar</i>	Menampilkan halaman portofolio sesuai kategori yang diinginkan seperti sertifikat, prestasi, dan lain-lain	Sesuai	Normal
Mahasiswa mengajukan data portofolio	Masukan data portofolio beserta bukti lalu klik 'tombol' kirim	Proses <i>upload</i> data ke <i>server</i> berjalan dengan cepat bahkan pada perangkat <i>mobile</i> proses berjalan kurang dari 3 detik	Sesuai	Normal
Admin Memeriksa daftar pengajuan	Membuka halaman daftar pengajuan	Melihat daftar pengajuan terbaru di halaman <i>dashboard</i>	Sesuai	Normal
Admin Melakukan validasi	Tekan tombol aksi lalu <i>input</i> -kan komentar dan pilih opsi terima atau tolak	Dilakukan pemeriksaan terhadap data ajuan lalu diberikan balasan diterima atau ditolak beserta dengan pesan deskripsinya	Sesuai	Normal

2. Lighthouse Testing

Lighthouse adalah ekstensi Google Chrome yang digunakan untuk menguji halaman *website* dan *progressive web apps*. Hasil pengujian menggunakan *lighthouse* dapat dilihat pada Gambar 13.

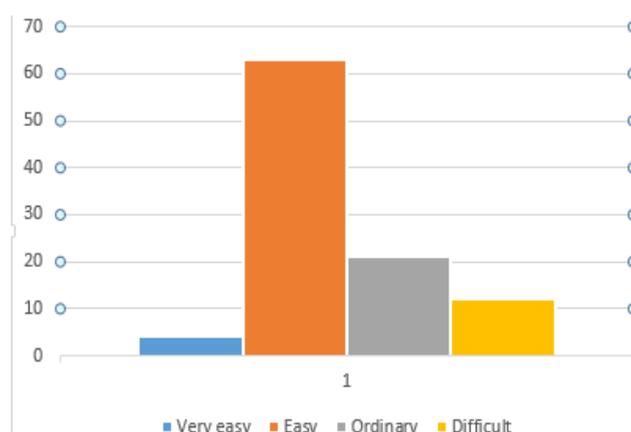


Gambar 13. Hasil *Lighthouse testing*.

Dari hasil *Lighthouse testing* sudah memenuhi kriteria *progressive web apps* dengan skor 100 dari 100, aplikasi juga sudah memenuhi kriteria pada indikator lainnya, indikator *performance* dengan skor 98 dari 100, indikator *accessibility* dengan skor 81 dari 100, indikator *best practice* dengan skor 100 dari 100, dan indikator *SEO* dengan skor 100 dari 100.

3. Usability Testing

Usability testing disebut juga pengujian kegunaan, ada beberapa indikator yang diujikan antara lain: *Learn Ability*, *Operability*, *Understandibility*, dan *Attractiveness*. *Usability testing* dilakukan dengan menyebarkan kuesioner kepada mahasiswa Prodi Teknik Informatika dan masyarakat umum, setelah dilakukan uji frekuensi, ditunjukkan hasil seperti pada Gambar 14.



Gambar 14. Hasil uji *usability*.

Dari hasil kuesioner yang dibagikan ke 100 responden tentang *usability testing* sistem sebanyak 4% menyatakan sangat mudah, 63% mudah, 21% biasa, dan 12% menyatakan sulit.

5. Kesimpulan

Berdasarkan hasil perancangan, pembangunan dan pengujian sistem maka dapat disimpulkan, Penggunaan *repository* E-Portofolio mahasiswa cukup efektif dalam mengelola portofolio mahasiswa, E-Portofolio mahasiswa juga dapat dikelola secara mandiri oleh mahasiswa dan dapat dilihat hasilnya oleh publik. Implementasi *progressive web apps* pada *repository* E-Portofolio mahasiswa juga berjalan dengan baik sehingga E-Portofolio tetap dapat diakses secara *offline* dan berjalan secara *responsive* pada perangkat komputer, laptop atau *mobile*.

Dari hasil *black box testing* menunjukkan semua fitur aplikasi berjalan dengan normal, Hasil *lighthouse testing* menunjukkan aplikasi berjalan dengan baik pada setiap indikator pengujian dengan skor di atas 80 dari 100, dan hasil *usability testing* menunjukkan aplikasi mudah digunakan oleh *user*.

Daftar Pustaka

- [1] D. Fernando, A. Anharudin, and F. Fadli, "Rancang Bangun Aplikasi E-Portofolio Hasil Karya Mahasiswa Unsera Menggunakan Metode Scrum," *JSiI (Jurnal Sist. Informasi)*, 2018, doi: 10.30656/jsii.v5i1.579.
- [2] R. Safitri, M. S. Gemilang, and D. Hermawan, "Repository Portofolio TA Mahasiswa Teknik Informatika UAI," *J. Al-Azhar Indones. Seri Sains Dan Teknol.*, 2018, doi: 10.36722/sst.v4i1.243.
- [3] A. R. Putra, "Rancang Bangun Sistem Informasi E-Portofolio Dosen," pp. 170–177, 2018.
- [4] W. D. Astika, A. H. Brata, and K. C. Brata, "Perancangan Aplikasi Asistensi Skripsi Mahasiswa berbasis Progressive Web App dengan Metode Prototyping," vol. 3, no. 11, 2019.
- [5] I. Arnomo, "Perbandingan Perangkat Lunak Repository Institusi : Studi Kasus pada Repository Institusi di Indonesia," *J. Inf.*, 2018, doi: 10.25139/ojsinf.v3i1.674.
- [6] S. Firmansyah, E. Chandra, and I. Aripin, "Pengembangan electronic portfolio (e-portfolio) sebagai a sssessment pembelajaran biologi," *J. Bio Educ.*, 2019.
- [7] N. Rahmawati and H. Mulyono, "Panduan Mudah Belajar Framework Laravel," *PT Alex Media Komputindo*, 2018.
- [8] L. Adi, R. J. Akbar, and W. N. Khotimah, "Platform e-Learning untuk Pembelajaran Pemrograman Web Menggunakan Konsep Progressive Web Apps," *J. Tek. ITS*, 2018, doi: 10.12962/j23373539.v6i2.24291.
- [9] A. A. Kurniawan, "Analisis Performa Progressive Web Application (Pwa) Pada Perangkat Mobile," *J. Ilm. Inform. Komput.*, 2020, doi: 10.35760/ik.2020.v25i1.2510.
- [10] A. Rahmatulloh, A. N. Rachman, and F. Anwar, "Implementasi Web Push Notification pada Sistem Informasi Manajemen Arsip Menggunakan PUSHJS," *J. Teknol. Inf. dan Ilmu Komput.*, 2019, doi: 10.25126/jtiik.201963936.
- [11] M. Rahmayu, "Rancang Bangun Sistem Informasi Pada Rumah Sakit Dengan Layanan Intranet Menggunakan Metode Waterfall," *J. Evolusi*, 2016.
- [12] J. Dermawan and S. Hartini, "Implementasi Model Waterfall pada Pengembangan Sistem Informasi Perhitungan Nilai Mata Pelajaran Berbasis Web pada Sekolah Dasar Al-Azhar Syifa Budi Jatibening," *Paradigma*, 2017.