

**RANCANG BANGUN SISTEM INFORMASI MANAJEMEN MAGANG
BERBASIS WEBSITE ODAMA STUDIO**



**Disusun sebagai salah satu syarat memperoleh Gelar Strata I
Pada Jurusan Informatika Fakultas Komunikasi dan Informatika**

**Oleh:
GENTUR WASKITA
L200170085**

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2021**

HALAMAN PERSETUJUAN

**RANCANG BANGUN SISTEM INFORMASI MANAJEMEN MAGANG
BERBASIS WEBSITE ODAMA STUDIO**

PUBLIKASI ILMIAH

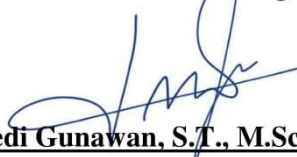
Oleh :

GENTUR WASKITA

L200170085

Telah diperiksa dan disetujui untuk diuji oleh :

Dosen Pembimbing



Dedi Gunawan, S.T., M.Sc, Ph.D

NIK. 1305

HALAMAN PENGESAHAN

**RANCANG BANGUN SISTEM INFORMASI MANAJEMEN MAGANG
BERBASIS WEBSITE ODAMA STUDIO**

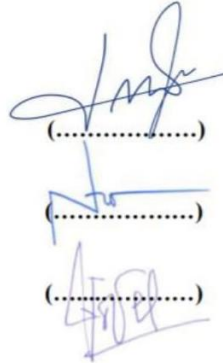
OLEH
GENTUR WASKITA
L200170085

Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Sabtu, 9 Oktober 2021
dan dinyatakan telah memenuhi syarat

Dewan Penguji :

1. **Dedi Gunawan, S.T., M.Sc, Ph.D**
(Ketua Dewan Penguji)
2. **Nurgiyatna, M.Sc., Ph.D.**
(Anggota I Dewan Penguji)
3. **Maryam, M.Eng.**
(Anggota II Dewan Penguji)

(.....)
(.....)
(.....)



Dekan
Fakultas Komunikasi dan Informatika



Nurgiyatna, M.Sc, Ph.D

NIK. 881

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam publikasi ilmiah ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 9 Oktober 2021

Penulis



GENTUR WASKITA

L200170085

RANCANG BANGUN SISTEM INFORMASI MANAJEMEN MAGANG BERBASIS WEB ODAMA STUDIO

Abstrak

Program magang merupakan suatu wadah bagi mahasiswa untuk mengenal situasi lingkungan dunia kerja nyata, dimana pada saat melaksanakan magang mahasiswa dapat melihat serta mendapat arahan langsung dan mampu beradaptasi dalam dunia kerja nantinya setelah menyelesaikan pendidikan. Dilain pihak, bagi instansi / perusahaan dengan adanya program magang dapat memantau kualitas kinerja yang dimiliki oleh calon SDM. Dalam proses pelaksanaannya pembimbing memberikan *jobdesk* kepada pemegang dalam bentuk project atau diluar project setelah diselesaikan pemegang langsung diserahkan ke pembimbing, tidak adanya sistem yang mencatat kinerja apa saja yang sudah diselesaikan oleh pemegang. Berdasarkan hal tersebut diperlukannya sistem yang mampu mencatat setiap kinerja dari pemegang sesuai *jobdesk* yang diberikan pembimbing. Proses rancang bangun sistem ini menggunakan metode pengembangan *Waterfall Model*, dengan Bahasa pemrograman yang digunakan *HTML*, *CSS*, dan *PHP* serta *MySQL* sebagai manajemen basis data. Hasil dari penelitian merupakan suatu sistem yang bisa meningkatkan efektifitas pembimbing dan pemegang dalam proses pelaksanaan program magang. Setiap kinerja pemegang akan tercatat sehingga memudahkan pembimbing dalam melakukan evaluasi kinerja tersebut dan membantu meningkatkan produktivitas pemegang maupun pembimbing dengan fitur yang dimiliki sistem.

Kata Kunci : *Kinerja, Magang, Sistem Informasi, Waterfall Model*

Abstract

The internship program is a place for students to get to know the real world of work environment, where when carrying out internships students can see and receive direct direction and be able to adapt to the world of work later after completing their education. On the other hand, agencies / companies with an internship program can monitor the quality of performance possessed by prospective HR. In the process of implementation, the supervisor gives a *jobdesk* to the intern either in the form of a project or outside the project after the intern is completed, it is directly submitted to the supervisor, there is no system that records any performance that has been completed by the intern. Based on this, a system is needed that is able to record every performance of the intern according to the *jobdesk* given by the supervisor. The design process of this system uses the *Waterfall Model* development method, with the programming language used *HTML*, *CSS*, and *PHP* and *MySQL* as database management. The results of this study are expected to be a system that can increase the effectiveness of supervisors and apprentices in the process of implementing the internship program. Each intern's performance will be recorded to make it easier for the supervisor to evaluate the performance and help increase the productivity of the intern and supervisor with the features of the system.

Keyword : Performance, Internship, Information System, Waterfall Model

1. PENDAHULUAN

Program praktik kerja atau magang memiliki tujuan utama yaitu mengembangkan ilmu yang didapatkan pemegang di bangku perkuliahan atau SMK/SMA setingkat, melalui peningkatan

kualitas, wawasan, dan pengalaman praktis didunia kerja nyata (Ratnasari & Lieli, 2017). Ilmu pengetahuan dan teknologi yang diberikan bangku sekolah kepada para lulusannya adalah bekal *hard skills*. Sementara itu, bekal *soft skills* diberikan melalui pengembangan kemampuan berkomunikasi baik secara lisan, tulisan maupun gambar, kemampuan bekerja secara mandiri atau dalam sebuah tim, kemampuan berlogika dan kemampuan menganalisis (Suharyanti et al., 2015). Program magang mempunyai peranan penting bagi dunia pendidikan ataupun instansi/perusahaan. Dimana bagi dunia pendidikan khususnya pemegang kegiatan tersebut memberikan manfaat berupa dapat merasakan iklim dunia kerja nyata, meningkatkan kompetensi atau *skill* sesuai dengan kemampuan yang dimiliki (Samidjo, 2017), selain itu mendapat kesempatan untuk bekerja di instansi/perusahaan tersebut, dan menambah hubungan profesional karena sering berinteraksi dengan para pekerja professional lain. Sedangkan bagi pihak instansi/perusahaan dengan adanya program ini sedikit banyaknya membantu pekerjaan karena mendapatkan tenaga kerja tambahan walaupun bersifat sementara (Pratama et al., 2017), selain itu dapat memantau kinerja yang dimiliki calon SDM sesuai dengan kebutuhan dari perusahaan dengan bekerja dibawah arahan atau *jobdesk* yang diberikan pembimbing ahli.

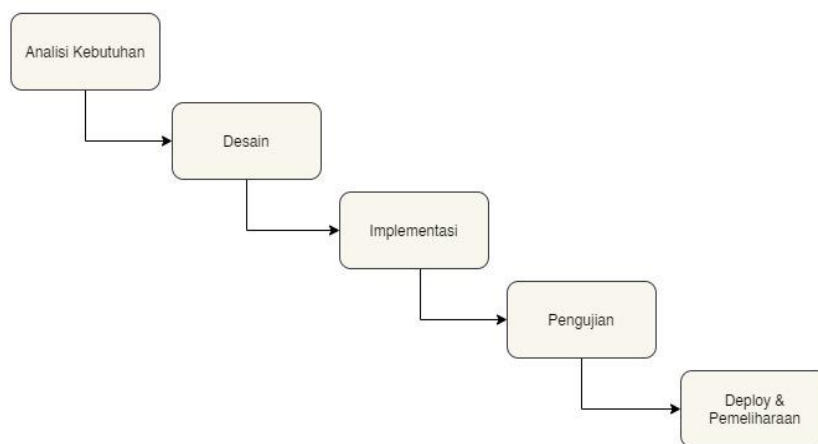
Odama Studio merupakan *creative studio* yang berfokus pada bidang desain *UI/UX* berbasis aplikasi maupun *website*, meningkatkan sebuah nilai bisnis atau *start-up* melalui optimasi merek (*Branding*). Pada pelaksanaan magang di Odama Studio pembimbing memberikan pekerjaan kepada pemegang dalam bentuk project beserta detail project maupun pekerjaan diluar project seperti input data ataupun pekerjaan yang tidak berhubungan dengan ilmu desain, setelah berhasil diselesaikan langsung dilaporkan ke pembimbing tidak ada pencatatan terhadap kinerja-kinerja yang berhasil diselesaikan. Berdasarkan permasalahan tersebut dibangunlah sistem informasi manajemen magang berbasis *website*, dengan sistem ini pemegang dapat memasukkan setiap kinerja yang berhasil diselesaikan dan pembimbing bisa melakukan perakapan, evaluasi terhadap kinerja tersebut. Dari rekap kinerja tersebut bisa membantu pemegang dalam membuat laporan magang apabila diperlukan oleh instansi asal pemegang. Pembimbing juga bisa membuat sebuah wadah atau bisa disebut *board* untuk project yang berisi detail project meliputi *jobdesk-jobdesk* project, komentar, *file* pendukung, team project dan aktivitas project. Dari wadah tersebut pemegang yang tergabung dalam project bisa menyelesaikan *jobdesk-jobdesk* yang sudah ditentukan. Pemegang juga wajib melakukan revisi terhadap kinerja yang di tolak oleh pembimbing, pembimbing dalam menolak kinerja akan diberikan sebuah catatan sebagai masukkan. Setelah pemegang berhasil

menyelesaikan magang, maka pembimbing akan mengupload sertifikat sebagai bukti sudah lulus magang yang bisa didownload oleh pemegang.

Dalam merancang bangun sistem ini seluruh *tools* yang digunakan *open source* antara lain *Webserver* yaitu *XAMPP*, *framework Bootstrap* (HTML & CSS) *Framework PHP Codeigniter 3* (PHP 7+), *Database Management System MySQL*, plugin *mPDF*, dan menggunakan metode pengembangan *Software Development Life Cycle* (SDLC) yaitu *Waterfall Model*. Penulis memilih model ini dikarenakan tahapan pada model mudah dipahami serta mudah dalam penerapan pengembangan sistem (Alshamrani & Bahattab, 2015). Dengan adanya sistem ini diharapkan proses pelaksanaan magang menjadi lebih efektif dan terorganisir antara pembimbing dan pemegang, apalagi dengan kondisi pandemic seperti ini yang mewajibkan kita untuk bekerja dari rumah sistem ini diharapkan bisa menjaga koordinasi antara pembimbing dan pemegang.

2. METODE

Proses rancang bangun sistem informasi manajemen magang melalui tahapan yang mengacu pada metode *System Development Life Cycle* (SDLC) *Waterfall Model*. Tahapan tersebut meliputi Analisis Kebutuhan, Desain Sistem, Implementasi / Pengkodean, Pengujian, Evaluasi dan Pemeliharaan (Royce, 1970). Gambar 1 menunjukkan tahapan *Waterfall Model*.



Gambar 1. Waterfall Model (Royce, 1970)

Pemilihan model *Waterfall* dikarenakan lebih mudah dipahami serta dalam implementasinya harus secara berurutan atau sekuensial, dengan kata lain setiap tahapan dalam metode harus benar-benar diselesaikan sebelum mengerjakan tahap selanjutnya. Keuntungan dari model ini ialah dapat dilakukan analisis dan kontrol terhadap tahapan-tahapan model, sehingga meminimalisir terjadinya kesalahan (Gunanto & Sudarmilah, 2020).

2.1. Analisis kebutuhan

Analisis Kebutuhan merupakan tahapan pertama dari model *Waterfall*, pada tahapan ini dilakukan pengumpulan data dan informasi, metode pengumpulan data dilakukan dengan cara observasi dan wawancara langsung terhadap *founder* dari Odama serta pengalaman penulis dalam menjalani program magang. Dari hasil observasi tersebut diperlukan sebuah sistem yang memiliki 3 jenis user yaitu Administrator, Pembimbing, dan Pemegang dimana masing-masing user memiliki *Accessibility* sesuai dengan level user.

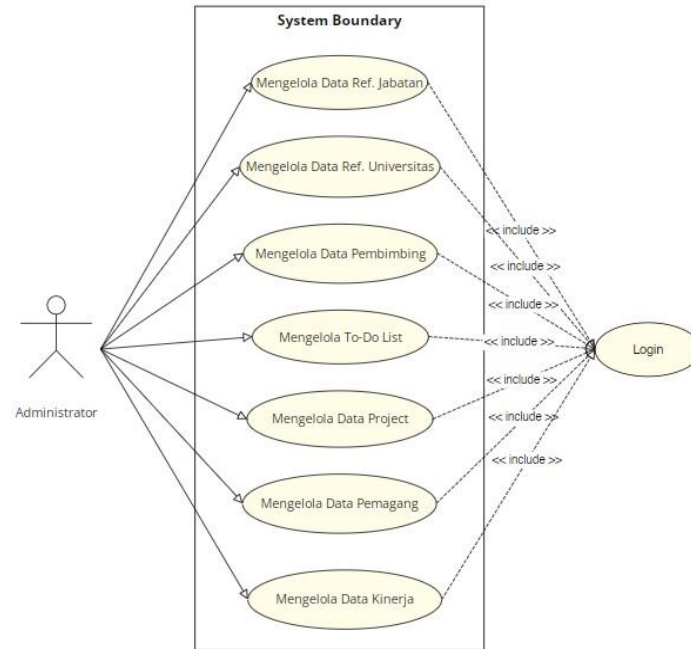
- 2.1.1. Administrator : *login* dan *logout* sistem, mengelola *To-Do List* guna membantu produktivitas administrator, membarui *profile* administrator, mengelola data pembimbing (menambah, mengubah, menghapus), mencetak kinerja pemegang, mengubah dan menghapus *profile* pemegang, menampilkan dan mengelola data *project*.
- 2.1.2. Pembimbing : *login* dan *logout* sistem, mengelola *To-Do List* guna membantu produktivitas pembimbing, membarui *profile* pembimbing, mengubah *password*, menampilkan dan menambahkan *profile* pemegang meliputi data diri dan *account*, mengubah *profile* pemegang, menghapus data pemegang, mengevaluasi kinerja pemegang, mencetak kinerja pemegang dalam bentuk *PDF*, menghapus kinerja pemegang, mengubah status pemegang dan mengupload sertifikat kelulusan magang, mengelola *board project*, mengelola *jobdesk* pada *project*, mengelola *team* pada *project*, mengelola *file* lampiran pada *project*.
- 2.1.3. Pemegang : *login* dan *logout* sistem, mengelola *To-Do List* guna membantu produktivitas pemegang, membarui *profile* pemegang, mengubah *password*, mengelola kinerja (menambahkan, mengubah, dan menghapus), melakukan revisi pada kinerja yang ditolak, menyelesaikan *jobdesk project*, mendownload sertifikat kelulusan magang.

2.2. Design

Tahapan desain berfokus pada perancangan mulai dari *Use Case Diagram*, *Activity Diagram*, *ER Diagram*, sampai pembuatan desain *Mockup* dari sistem yang akan dibangun. Tujuan dari Desain Sistem ini adalah menguraikan pekerjaan apa saja yang harus dilakukan agar pada saat proses implementasi atau penulisan kode program lebih mudah, karena sistem yang akan dibangun sudah ada gambaran jelas. Berikut merupakan rancangan dari sistem informasi manajemen magang.

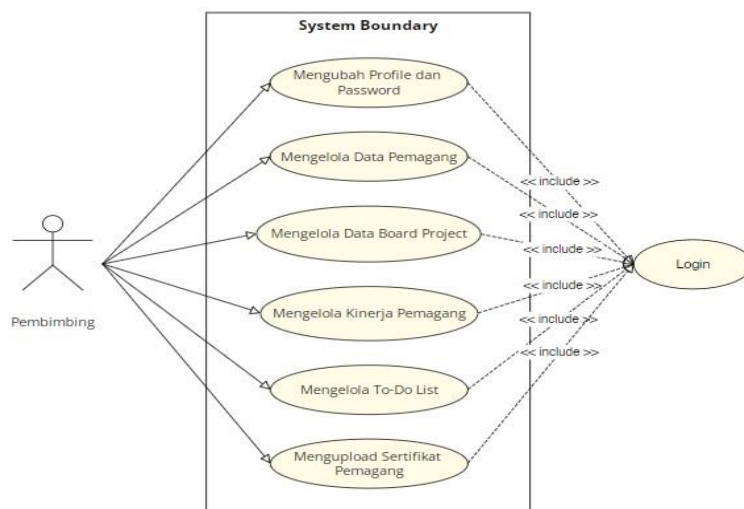
2.2.1. Use Case Diagram

Use Case Diagram menjabarkan mengenai setiap interaksi masing-masing user atau tindakan apa saja yang dapat dilakukan terhadap sistem sesuai dengan level user tersebut.



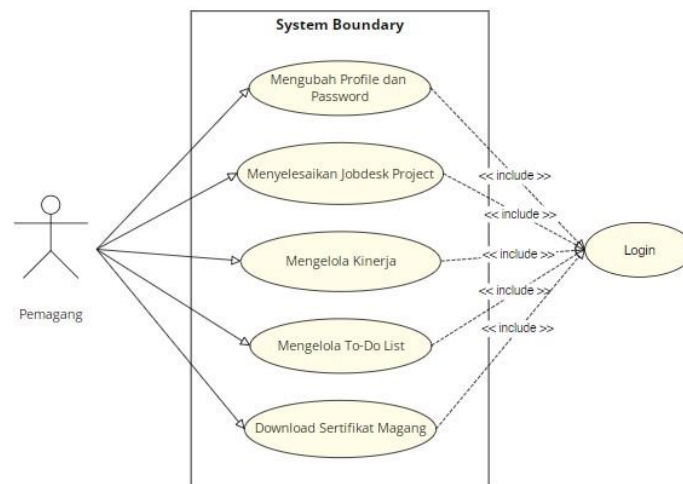
Gambar 2. Use Case Administrator

Gambar 2 menjelaskan apa saja tindakan yang bisa dilakukan oleh user Administrator, tetapi untuk melakukan tindakan-tindakan tersebut diwajibkan untuk melakukan autentifikasi / *login* terlebih dahulu. Proses menambahkan data Pemegang hanya bisa dilakukan oleh user Pembimbing, sedangkan untuk menambahkan data Pembimbing baru hanya bisa dilakukan oleh user Administrator.



Gambar 3. Use Case Pembimbing

Gambar 3 merupakan tindakan-tindakan yang bisa dilakukan oleh user Pembimbing dengan melakukan autentifikasi terlebih dahulu. Pada user Pembimbing memiliki hak untuk menambahkan data Pemegang baru, mengupload sertifikat kelulusan bukti lulus magang. Setelah berhasil menambahkan *board project* baru, Pembimbing bisa menambahkan *jobdesk* pada *board* tersebut beserta detail lain seperti melampirkan *file*, menambah *team*, melakukan interaksi dengan *team* melalui fitur *comment*.

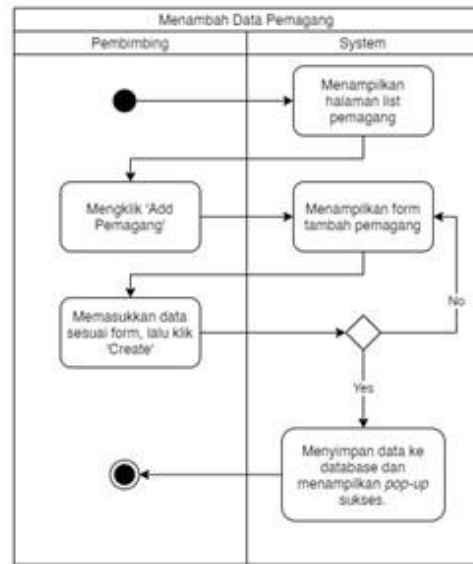
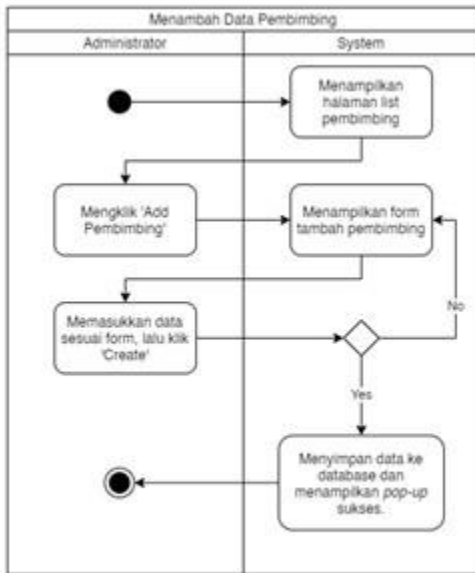


Gambar 4. Use Case Pemegang

Gambar 4 merupakan aktivitas yang dimiliki oleh user Pemegang, seperti user yang lain diwajibkan untuk *login* terlebih dahulu untuk melakukan aktivitas tersebut. Setelah user Pembimbing membuat *board project* dan menambahkan detail-detail *project*, user Pemegang yang tergabung bisa melakukan aktivitas seperti menyelesaikan *jobdes*, mendownload *file* yang diperlukan, berinteraksi dengan anggota lain yang tergabung, dan melihat aktivitas yang ada di *board project* tersebut.

2.2.2. Activity Diagram

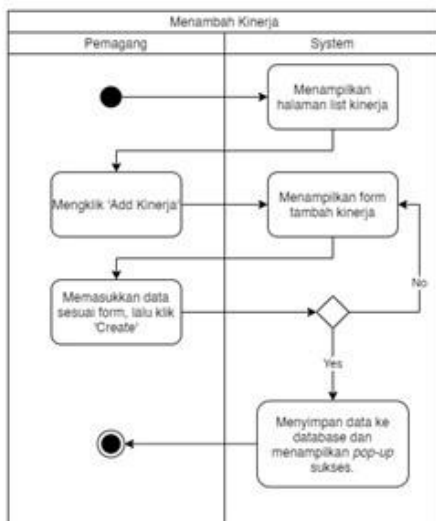
Activity Diagram merupakan diagram yang menggambarkan proses interaksi secara vertikal antara user dengan sistem, atau dapat diartikan menjabarkan dengan baik alur kerja sistem berdasarkan interaksi. Pada gambar 5 menjelaskan urutan dari proses tersebut, mulai dari mengakses halaman data Pembimbing, lalu klik tombol 'Add Pimbimbing' maka akan diarahkan ke halaman tambah pembimbing yang berisi *form* wajib untuk diisi sesuai ketentuan, setelah berhasil mengisi maka sistem akan menyimpan data tersebut.



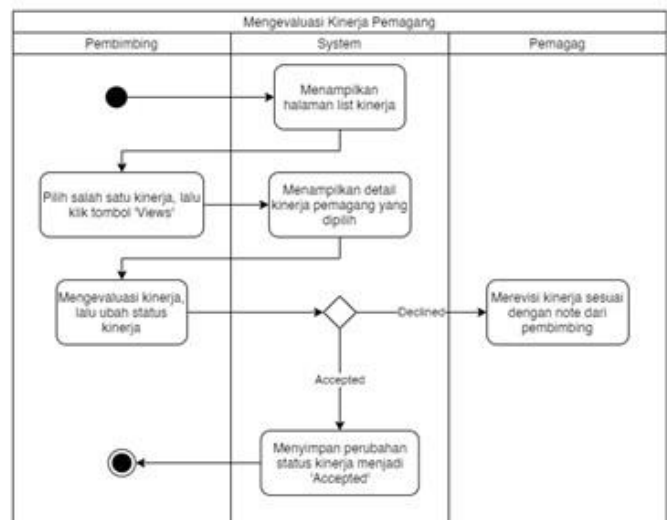
Gambar 5 Activity Diagram Menambahkan Data Pembimbing 6 Activity Diagram Pembimbing Menambahkan Data Pemegang

User Pembimbing berhasil ditambahkan, user tersebut berhak untuk menambahkan data Pemegang baru melalui halaman list data Pemegang tetapi dengan *level* akses user yaitu Pembimbing. Untuk proses menambahkan data Pemegang pada Gambar 6 sama seperti saat Administrator menambahkan user Pembimbing baru, dengan Pembimbing mengakses halaman data Pemegang, lalu menuju kehalaman tambah data yang berisi form wajib untuk diisi setelah berhasil divaliasi oleh sistem maka akan diarahkan kembali ke halaman list data Pemegang.

Pada gambar 7 merupakan gambar alur user Pemegang menambahkan kinerja yang berhasil diselesaikan untuk dievaluasi oleh Pembimbing. dengan user mengakses halaman list kinerja lalu klik tombol 'Add Kinerja' maka akan menampilkan *modal form* yang wajib diisi, setelah berhasil mengisi data inputan akan divalidasi dan apabila sesuai maka sistem akan menyimpan data kinerja tersebut.



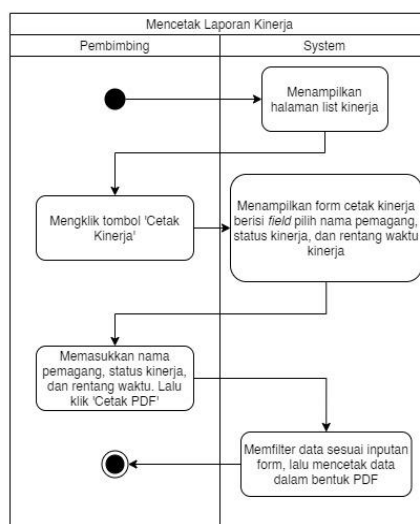
Gambar 7 Pemegang Menambahkan Kinerja



Gambar 8 Pembimbing Mengevaluasi Kinerja Pemegang

Data kinerja yang berhasil masuk akan di evaluasi oleh Pembimbing. Pada gambar 8 merupakan alur dari saat Pembimbing melakukan evaluasi dan Pemegang akan melakukan revisi terhadap kinerja tersebut.

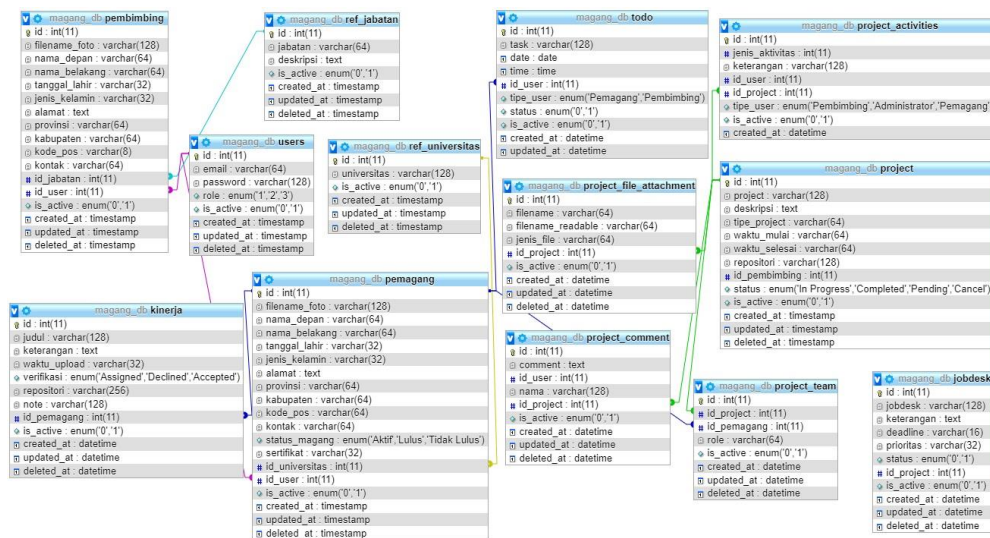
Pada gambar 9 merupakan alur bagi user Pembimbing dalam mencetak laporan kinerja dari pemegang. Mulai dari Pembimbing mengakses halaman kinerja, lalu klik ‘Cetak Kinerja’ maka akan tampil *model form* yang berisi inputan pilihan kinerja pemegang siapa yang akan dicetak, status kinerja tersebut. Untuk status bersifat opsional sehingga apabila hanya memilih pemegang akan dicetak semua kinerja pemegang tersebut dengan mengabaikan status kinerja pemegang.



Gambar 9. Activity Diagram Pembimbing Mencetak Laporan Kinerja

2.2.3. Entity Relationship Diagram

ER Diagram merupakan proses penggambaran rancangan basis data yang meliputi relasi antara entitas atau objek beserta atribut-atribut. Pada tahap ini proses perancangan *ER Diagram* bertujuan untuk memberi kemudahan dalam proses pembuatan basis data berdasarkan rancangan telah ditentukan.

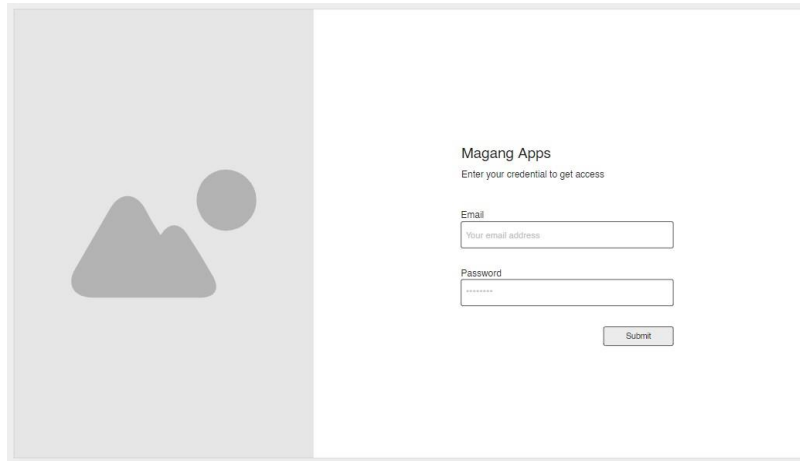


Gambar 10. ER Diagram System Informasi Magang

Pada *ER Diagram* diatas terdapat tabel *project* yang berelasi dengan tabel *pembimbing*, yang mana *project* hanya bisa dibuat oleh user Pembimbing saja sehingga relasi antara tabel *pembimbing* dan *project* ialah *one to many* disini Pembimbing sebagai inisiasi *project*, sedangkan antara tabel *pemegang* dan *project* ialah *many to many* dimana setiap user bisa join lebih dari 1 *project* dan 1 *project* bisa di join banyak pemegang.

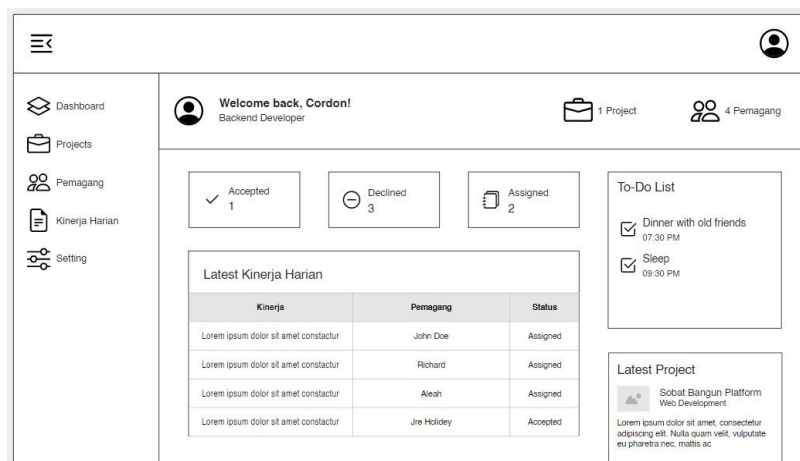
2.2.4. Mockup

Pembuatan desain *mockup* atau *wireframe* bertujuan sebagai acuan dalam proses pembuatan interface pada program. Gambar 11 rancangan *mockup* dari sistem *login* user yang merupakan langkah awal apabila ingin mengakses halaman utama user. Dengan memasukkan *email* dan *password* sistem akan mengautentifikasi apakah data yang dimasukkan cocok dengan level user tertentu dan jika salah akan tetap berada dihalaman *login* sistem.



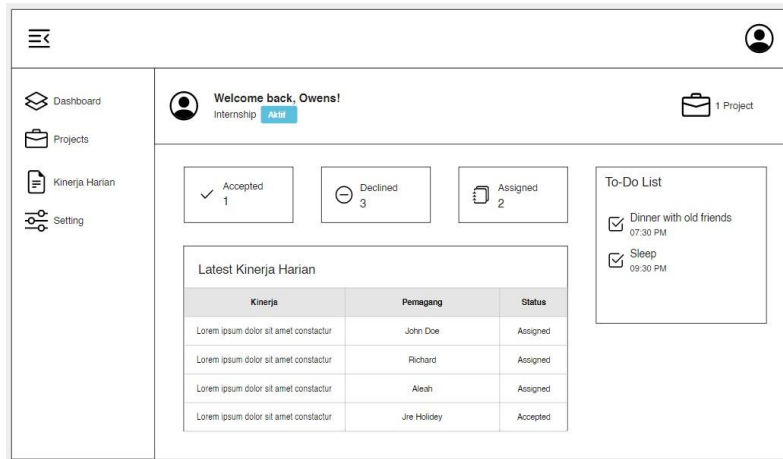
Gambar 11. Mockup Login User

Gambar 12 merupakan rancangan *mockup* dari *Dashboard* Pembimbing, apabila user berhasil *login* dan jenis akun teridentifikasi sebagai Pembimbing maka akan diarahkan ke halaman *Dashboard* Pembimbing. Pada *sidebar* terdapat menu-menu yang disesuaikan dengan kebutuhan user Pembimbing meliputi *Dashboard*, *Projects*, *Pemagang*, *Kinerja Harian*, dan *Setting Profile*.



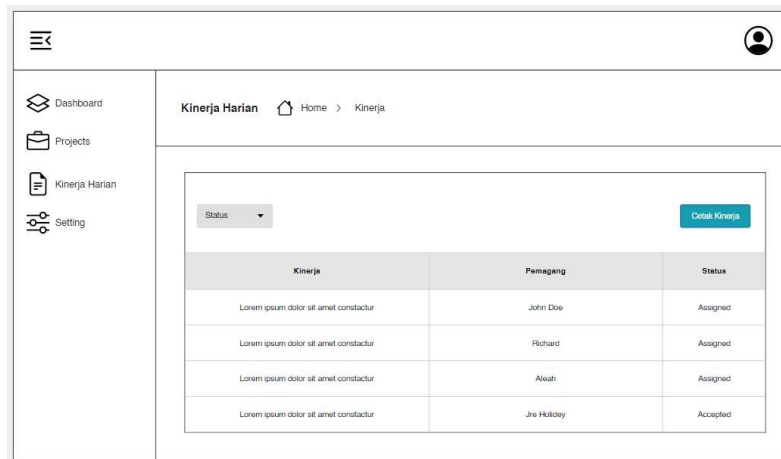
Gambar 12. Mockup Dashboard Pembimbing

Rancangan *mockup Dashboard* user Pemagang dapat dilihat pada gambar 13, yang mana pada *Dashboard* terdapat ringkasan berapa jumlah kinerja yang dimasukkan, kinerja yang berhasil disetujui, dan kinerja yang ditolak oleh pembimbing. Sedangkan untuk menu sidebar dirancang sesuai dengan kebutuhan user Pemagang



Gambar 13. Mockup Dashboard Pemegang

Gambar 14 merupakan rancangan *mockup* halaman kinerja pemegang, apabila user telah menyelesaikan sebuah kinerja dan ingin mengunggah kinerja tersebut untuk dievaluasi oleh pembimbing maka akan mengakses halaman kinerja. User Pemegang jika memiliki hak akses untuk mencetak laporannya.



Gambar 14. Mockup Halaman Kinerja Pemegang

2.3.Implementasi

Berdasarkan hasil dari desain sistem yang meliputi penentuan *Use Case*, *Activity Diagram*, *ER Diagram*, dan *Mockup*. Tahapan selanjutnya ialah pengkodean untuk mengimplementasikan rancangan tersebut menjadi sebuah sistem terstruktur. *Tools software* yang digunakan meliputi *Framework Codeigniter 3 (PHP)*, *Bootstrap (HTML & CSS)*, dan *MySQL* sebagai *Database Management System*. Alasan menggunakan bantuan *framework* ialah mempercepat pengembangan sistem dan struktur aplikasi yang terorganisir. *Framework Codeigniter 3* menggunakan *pattern* atau pola arsitektur *Model*, *View*, dan *Controller*. Pola tersebut memisahkan antara tampilan, data, dan proses.

2.3.1. *Controller*, Bertugas untuk mengendalikan data dan tampilan, yang nantinya akan dijalankan berdasarkan permintaan dari user. Namun, terkadang permintaan dari user tidak selalu memerlukan aksi dari model.

- a. *Auth.php*, merespon request dari user yang berhubungan dengan aktivitas *login* dan *logout* diantaranya memanggil data dari table user untuk dilakukan validasi sebagai syarat untuk mengakses *privileges* yang dimiliki user, maka *controller* *Auth* akan memproses sesuai dengan permintaan tersebut.
- b. *Dashboard.php*, *controller Dashboard* biasanya berisi fungsi untuk memanggil data apa saja yang akan ditampilkan ke halaman *Dashboard*.
- c. *Todo.php*, merespon request dari user yang berhubungan dengan pengelolaan data *To-Do List* yang meliputi menampilkan, menambah dan menghapus data.
- d. *Pembimbing.php*, berisi fungsi-fungsi merespon request dari user guna mengelola data *Pembimbing* diantaranya menampilkan, menambah, menghapus, dan mengubah data.
- e. *Pemagang.php*, berisi fungsi-fungsi merespon request dari user guna mengelola data *Pemagang* diantaranya menampilkan, menambah, menghapus, dan mengubah data.
- f. *Kinerja.php*, berisi fungsi-fungsi merespon request dari user guna mengelola data *Kinerja* diantaranya menampilkan, menambah, menghapus, dan mengubah data.
- g. *Project.php*, berisi fungsi-fungsi merespon request dari user guna mengelola data *Project* diantaranya menampilkan, menambah, menghapus, dan mengubah data.
- h. *Universitas.php*, berisi fungsi-fungsi merespon request dari user guna mengelola data *Universitas* diantaranya menampilkan, menambah, menghapus, dan mengubah data.
- i. *Roles.php*, berisi fungsi-fungsi merespon request dari user guna mengelola data *Roles* diantaranya menampilkan, menambah, menghapus, dan mengubah data.

2.3.2. *Model*, Bertugas untuk mengatur, menyiapkan, memanipulasi dan mengorganisasikan data dari basis data sesuai dengan instruksi dari *controller*.

- a. *PembimbingModel.php*, merespon permintaan *Controller* berhubungan dengan data *Pembimbing* yang *query* langsung ke tabel *Pembimbing*.
- b. *PemagangModel.php*, merespon permintaan *Controller* berhubungan dengan data *Pemagang* yang *query* langsung ke tabel *Pemagang*.
- c. *KinerjaModel.php*, merespon permintaan *Controller* berhubungan dengan data *Kinerja* yang *query* langsung ke tabel *Kinerja*.

- d. *ProjectModel.php*, merespon permintaan *Controller* berhubungan dengan data *Project* yang *query* langsung ke tabel *Project*.
- e. *UniversitasModel*, merespon permintaan *Controller* berhubungan dengan data *Universitas* yang *query* langsung ke tabel *Universitas*.
- f. *RolesModel*, merespon permintaan *Controller* berhubungan dengan manipulasi data *Roles* yang *query* langsung ke tabel *Roles*.

2.3.3. *View*, respon dari *controller* akan disajikan oleh *view* dalam bentuk informasi yang mudah dimengerti oleh user sesuai dengan instruksi yang diminta dari *controller*.

Sistem dikembangkan mengacu pada kemampuan dan kebutuhan masing-masing user yang sudah ditetapkan. Setelah proses implementasi desain sudah selesai, dilanjutkan dengan pengujian terhadap sistem.

2.4. Pengujian Sistem

Tahapan pengujian program dilakukan setelah sistem selesai dibuat, pengujian program dilakukan untuk melihat dan memastikan bahwa sistem sudah berjalan seperti yang dirancang. Pengujian sistem menggunakan metode *Blackbox Testing* dan pengujian *System Usability Scale*.

2.4.1. Blackbox Testing

Blackbox Testing dilakukan untuk menguji fitur dari sistem (Supriyono, 2020), apakah sudah berjalan seperti yang diharapkan atau masih terdapat kekurangan. *Blackbox Testing* bertujuan untuk mengevaluasi kembali apakah sistem yang dikembangkan masih terdapat *error* atau *bug* didalamnya. Jika dalam proses pengujian terdapat hal tersebut maka akan dilakukan perbaikan terhadap sistem.

2.4.2. System Usability Scale

Pengujian *SUS* dilakukan dengan cara membagikan kuisisioner kepada responden dengan daftar pernyataan yang sudah ditentukan (Kaban et al., 2019), yang bertujuan untuk melihat respon dari user serta mendapatkan penilaian terhadap kelayakan, kenyamanan user saat mengakses sistem, dan persepsi terhadap sistem tersebut.

Tabel 1. Daftar pertanyaan kuisisioner SUS (Brooke, 1995)

No	Daftar Pernyataan
1	Saya berpikir akan menggunakan sistem ini lagi.
2	Saya merasa sistem ini rumit untuk digunakan.
3	Saya merasa sistem ini mudah digunakan.
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.

6	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).
7	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan.
9	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.
10	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

Setiap pertanyaan diberikan pilihan skala 1 - 5 untuk dijawab berdasarkan pada seberapa banyak mereka setuju dengan setiap pernyataan tersebut terhadap fitur dari sistem. Nilai 1 berarti sangat tidak setuju dan 5 berarti sangat setuju dengan pernyataan tersebut. Setelah kalkulasi setiap pernyataan ganjil dan genap, selanjutnya adalah menjumlahkan nilai setiap pernyataan tersebut. Perlu diingat skor tertinggi untuk tiap pertanyaan adalah 4 dan yang terendah adalah 0. Setelah menjumlahkan skor 10 pernyataan tersebut, langkah selanjutnya adalah mengalikannya dengan 2,5 dan itu lah hasil *SUS Score* untuk fitur-fitur yang dimiliki sistem, sedangkan untuk skor tertingginya ialah 100 (Brooke, 1995).

Nantinya pada saat pengujian sistem akan dilakukan pada sebuah perangkat PC atau Laptop yang sudah terinstall *XAMPP* sebagai *local server*, pada tahap pengujian masih belum diperlukan *hosting website* karena masih sebatas pengujian. Hasil dari tahap pengujian nantinya dijadikan sebagai acuan untuk memperbaiki sistem dan penyempurnaan sistem sebelum benar-benar diimplementasi.

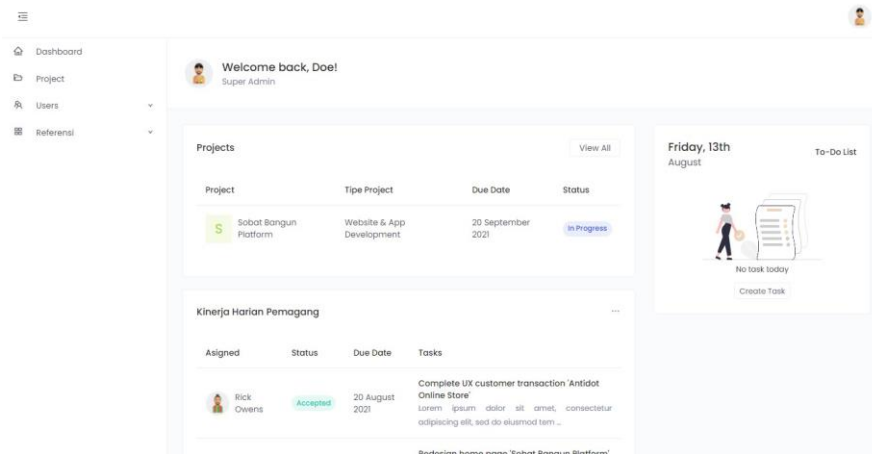
2.5. Deploy & Pemeliharaan

Tahapan terakhir yaitu *Deploy & Pemeliharaan*, pada tahapan ini sistem sudah diterapkan secara langsung atau di *deploy*. Agar dapat diakses melalui internet diperlukan layanan web hosting tentunya berbayar dengan mengutamakan keamanan, jasa penyedia layanan hosting berbayar seperti Niagahoster dan Hostinger Indonesia selanjutnya informasi dari *website* akan disimpan pada server web hosting. Setelah itu *website* sudah bisa diakses melalui *web browser* diantaranya *Google Chrome, Mozilla Firefox, Opera, Safari*, dll. Tidak menutup kemungkinan saat sistem sudah dapat diakses melalui internet tidak terjadi *error* atau bug, apabila ditemukan hal tersebut pada sistem maka akan dilakukan perbaikan terhadap sistem agar dapat berfungsi seperti yang seharusnya.

3. HASIL DAN PEMBAHASAN

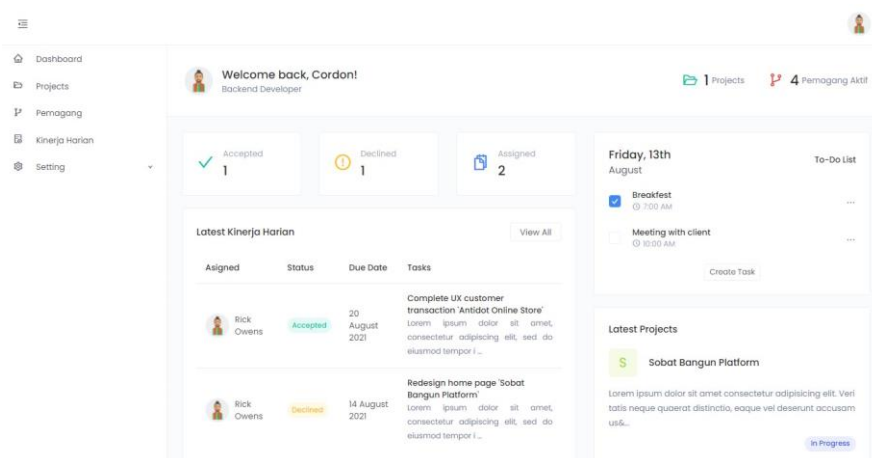
Proses rancang bangun sistem meliputi menganalisa kebutuhan sistem, mengimplementasi analisa ke sebuah rancangan, dan mengeksekusi rancangan tersebut telah menghasilkan sistem informasi manajemen magang berbasis *website* pada Odama Studio.

3.1. Hasil Penelitian



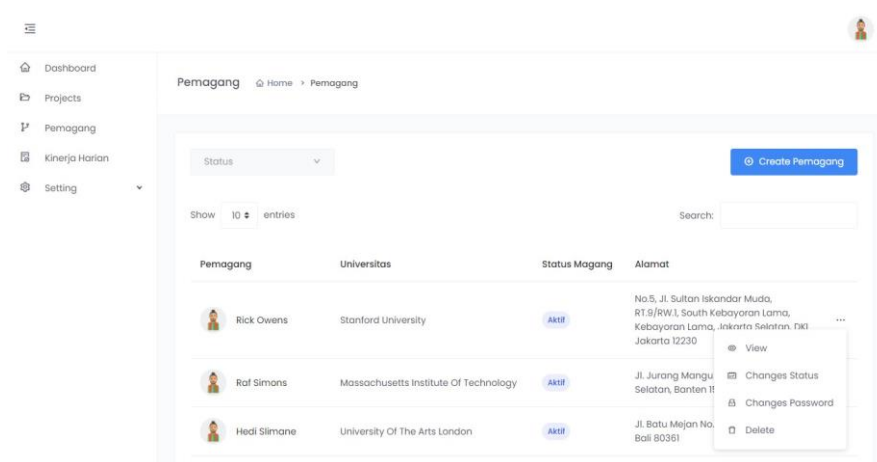
Gambar 15. User Interface Dashboard Administrator

Gambar 15 merupakan tampilan halaman *Dashboard* dari user Administrator, berisi informasi mengenai *board project* terbaru yang diinisiasi oleh user Pembimbing, terdapat juga kinerja terbaru dari Pemegang, dan terdapat fitur *To-Do List* untuk membantu Administrator mengatur jadwal hariannya guna meningkatkan produktivitas.



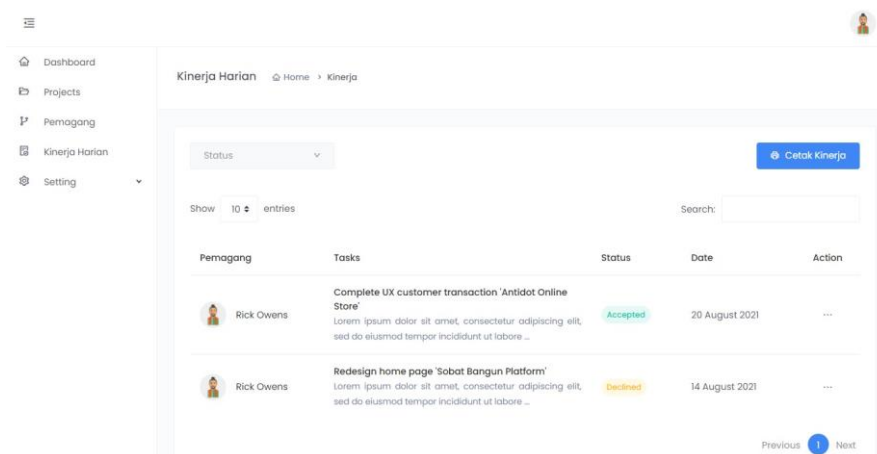
Gambar 16. User Interface Dashboard Pembimbing

Gambar 16 merupakan tampilan dari *Dashboard* user Pembimbing hampir sama dengan *Dashboard* milik Administrator, tetapi di sini terdapat informasi mengenai berapa jumlah *board project* yang telah dibuat oleh user Pembimbing tersebut, berapa jumlah pemegang aktif, dan ringkasan *project* terbaru.



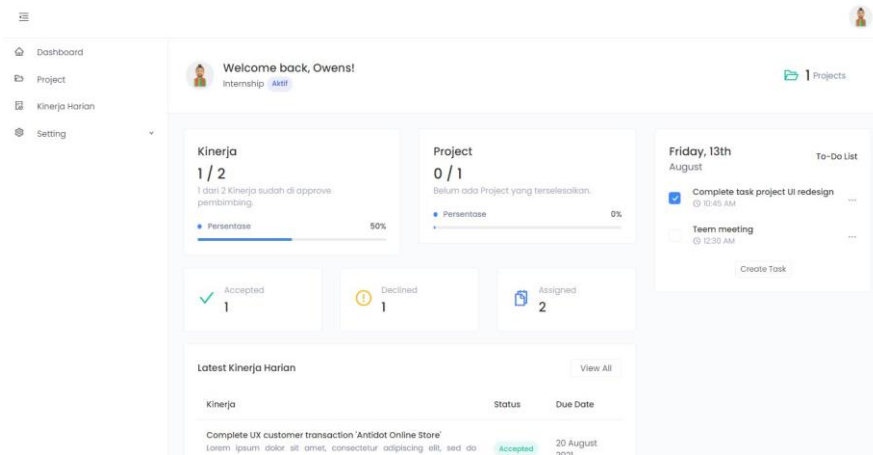
Gambar 17. User Interface Data Pemagang

Gambar 17 menampilkan halaman pemagang di sisi user Pembimbing. Setelah berhasil menambahkan Pemagang baru, terdapat menu aksi yaitu *view*, *changes status*, *changes password*, dan *delete*.



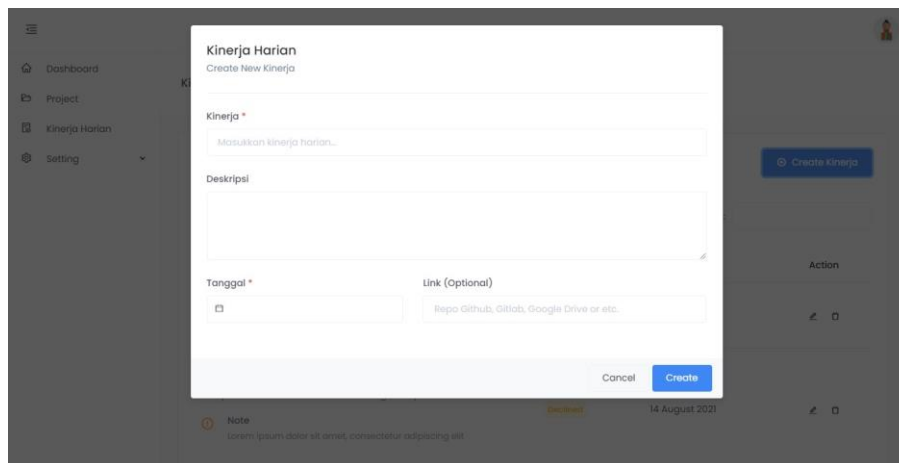
Gambar 18. User Interface Data Kinerja

Gambar 18 merupakan tampilan dari halaman data kinerja di sisi user Pembimbing. Setelah berhasil user Pemagang mengunggah kinerjanya maka akan tampil di halaman data kinerja dan akan dievaluasi oleh Pembimbing.



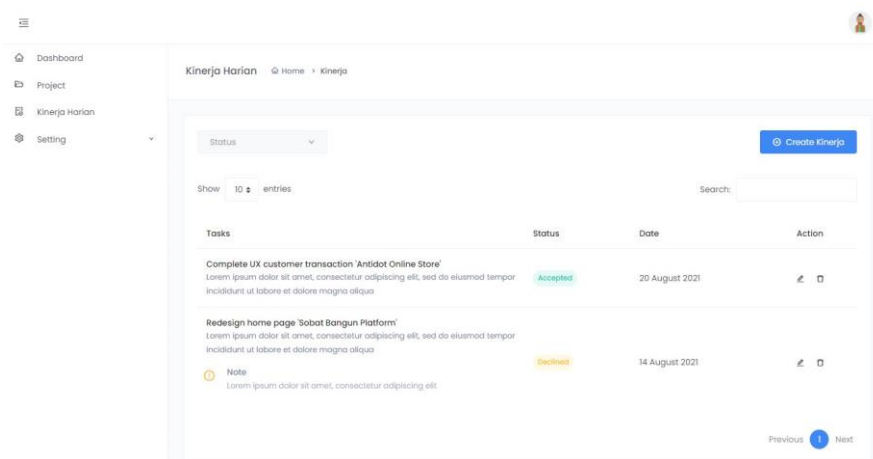
Gambar 19. User Interface Dashboard Pemangang

Gambar 19 merupakan tampilan dari *Dashboard* Pemangang. Sama seperti user yang lain, untuk user Pemangang terdapat informasi mengenai berapa jumlah kinerja Pemangang tersebut yang berhasil diunggah, berapa jumlah kinerja yang ditolak, dan berapa jumlah kinerja yang di terima.



Gambar 20. User Interface Modal Form

Gambar 20 merupakan tampilan halaman data kinerja di sisi user Pemangang saat akan menambahkan kinerja. Pada form terdapat inputan judul kinerja, deskripsi kinerja tersebut, tanggal berapa kinerja diselesaikan, dan link yang mengarah ke hasil kinerja tersebut bila memang ada.



Gambar 21 User Interface Kinerja Pemangang

Gambar 21 merupakan tampilan dari halaman kinerja di sisi user Pemangang. Terdapat informasi mengenai kinerja beserta deskripsi, status kinerja tersebut, kapan kinerja diselesaikan, dan aksi-aksi yang bisa dilakukan oleh Pemangang terhadap kinerja nya yaitu *Update* dan *Delete*. Pada kinerja Pemangang yang berstatus *Declined* akan menampilkan sebuah catatan yang berasal dari Pembimbing, sebagai catatan atau masukan terhadap kinerja tersebut untuk di revisi oleh Pemangang.

3.2. Pengujian *Black Box*

Merupakan metode pengujian terhadap sistem yang telah dibangun, untuk menemukan *Bug* atau *Error* pada sistem sebelum sistem benar-benar di implementasikan, tanpa harus memperhatikan detail sistem tersebut. Adanya pengujian sistem dapat mengetahui apakah sistem tersebut sudah memenuhi semua kebutuhan dari user. Pengujian ini berfokus pada input yang tersedia dan output yang diharapkan tanpa pengetahuan tentang internal programnya, jadi tidak perlu pengembang sistem untuk melakukan pengujian terhadap sistem cukup yang memahami secara garis besar sistem tersebut.

Tabel 2. Hasil Pengujian *Black Box*

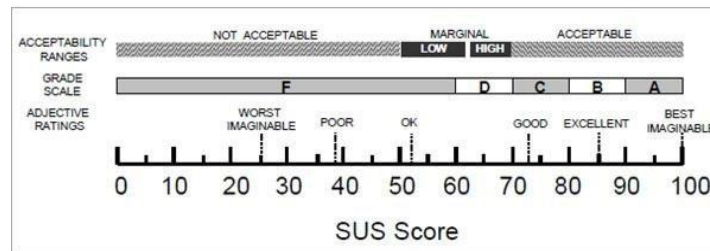
No.	Interaksi	Kondisi Pengujian	Output	Status
1	<i>Login</i>	Memasukkan <i>Email</i> dan <i>Password</i> yang valid.	Mengarah ke halaman <i>Dashboard</i> masing-masing user.	Valid
2	User mengubah <i>password</i>	Setelah berhasil <i>login</i> , klik <i>profile</i> pojok kanan atas, pilih 'Changes Password' akan muncul modal form yang berisi inputan <i>Password</i> dan <i>New Password</i> . form wajib untuk diisi apabila ingin mengubah <i>password</i> .	Menampilkan pesan <i>password</i> berhasil diperbarui.	Valid
3	Administrator mengelola (Tambah, Hapus, dan Edit) user	Mengakses halaman list data Pembimbing, Pembimbing melakukan penambahan, pengeditan, dan	Menampilkan pesan data berhasil diubah, ditambah, atau dihapus.	Valid

	Pembimbing.	penghapusan pada data Pembimbing.		
4	Pembimbing mengelola (Tambah, Hapus, dan Edit) user Pemegang.	Mengakses halaman list data Pemegang, Pembimbing melakukan penambahan, pengeditan, dan penghapusan pada data Pemegang.	Menampilkan pesan data berhasil diubah, ditambah, atau dihapus.	Valid
5	User Pembimbing dan Pemegang mengubah <i>profile</i> .	Setelah berhasil <i>login</i> sebagai Pembimbing atau Pemegang, pilih submenu ' <i>Profile</i> ' pada menu ' <i>Setting</i> ', mengarah ke halaman ' <i>Ubah Profile</i> ' dan mengisikan data sesuai dengan validasi.	Kembali ke <i>Dashboard</i> user setelah berhasil mengubah <i>profile</i> .	Valid
6	Pemegang mengelola (Tambah, Hapus, dan Edit) kinerja harian.	Mengakses halaman kinerja Pemegang, Pemegang melakukan penambahan, pengeditan, dan penghapusan pada data kinerja harian.	Menampilkan pesan data berhasil diubah, ditambah, atau dihapus.	Valid
7	User mengelola (Tambah, Hapus, dan Edit) <i>To-Do List</i> .	Mengakses halaman Dashoard, User melakukan penambahan, pengeditan, dan penghapusan pada <i>To-Do List</i> .	Menampilkan pesan data berhasil diubah, ditambah, atau dihapus.	Valid
8	Pembimbing mengelola (Tambah, Hapus, dan Edit) board project.	Mengakses halaman board project, User melakukan penambahan, pengeditan, dan penghapusan pada board project.	Menampilkan pesan data berhasil diubah, ditambah, atau dihapus.	Valid
9	Pembimbing mengevaluasi kinerja Pemegang	Mengakses halaman kinerja, pembimbing memilih kinerja pemegang lalu mengklik ' <i>Views</i> ', maka akan tampil modal berisi detail kinerja pemegang beserta form inputan berisi select status dan note. Apabila Pembimbing memilih status ' <i>Declined</i> ' maka harus mengisikan note untuk dijadikan revisi oleh pemegang.	Menampilkan pesan kinerja berhasil diperbarui.	Valid
10	Pembimbing mencetak kinerja pemegang.	Mengakses halaman kinerja, lalu klik ' <i>Cetak Kinerja</i> '. Terdapat form berisi nama pemegang, status kinerja, dan rentang waktu. Untuk inputan nama wajib diisi sebagai filter, pemegang siapa yang akan dicetak kinerjanya.	File PDF berisi kinerja pemegang yang dipilih.	Valid
11	Pembimbing mengunggah sertifikat kelulusan bukti magang.	Mengakses halaman pemegang, pilih action ' <i>Changes Status</i> '. Terdapat inputan berupa status pemegang dan upload file sertifikat.	Menampilkan pesan data berhasil diperbarui.	Valid
12	Pemegang menyelesaikan <i>jobdesk</i> yang ada pada project.	Mengakses halaman project, lalu tampil project apa saja yang berhasil tergabung. Mengklik salah satu project maka akan mengarah ke detail project, terdapat <i>jobdesk-jobdesk</i> apa saja yang perlu diselesaikan, lalu Pemegang centang salah satu <i>jobdesk</i> .	<i>Jobdesk</i> akan tercentang dan card aktivitas akan menampilkan siapa yang menyelesaikan project tersebut.	Valid

3.3.Pengujian SUS

Pengujian ini digunakan untuk mengukur fungsionalitas sistem menurut sudut pandang subyektif pengguna. Dengan membagikan kuisioner kepada pengguna yang terdiri dari 10 jenis pertanyaan seperti pada table 1, setiap pertanyaan memiliki 5 pilihan jawaban meliputi '*Sangat Tidak Setuju*', '*Tidak Setuju*', '*Ragu-ragu*', '*Setuju*', dan '*Sangat Setuju*'. Dari pilihan jawaban tersebut memiliki nilai angka skala 1 – 5, pada setiap pertanyaan memiliki

perhitungan yang berbeda yaitu pertanyaan bernomor ganjil, nilai dihitung dengan cara nilai pertanyaan dikurangi 1. Sedangkan pertanyaan bernomor genap nilai akhir didapat dari nilai 5 dikurangi nilai pertanyaan yang didapat dari pengguna.



Gambar 22. SUS Score

Setelah didapatkan data respon dari pengguna, data tersebut akan dikalkulasi untuk memperoleh nilai *SUS*. Gambar 22 memperlihatkan kategori nilai *SUS*, nilai akan diinterpretasikan menggunakan kategori penerimaan (*Acceptability*), skala nilai (*Grade Scale*), dan *Adjectives Ratings*. Minimal nilai *SUS* atau rata-rata yang harus diperoleh sistem agar dapat dikategorikan sebagai sistem yang layak ialah sebesar 68 (Brooke, 2013).

Tabel 3. SUS Testing Result

Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Jumlah	Skor SUS (2,5 x Jumlah)
1	4	3	3	2	4	4	3	4	3	2	32	80
2	3	3	3	3	3	3	3	4	3	3	31	77,5
3	4	4	3	3	3	4	3	2	2	3	31	77,5
4	4	3	4	3	4	3	4	3	4	3	35	87,5
5	2	3	3	3	3	4	3	4	3	4	33	82,5
6	4	4	3	3	4	4	3	3	4	2	34	85
7	4	4	4	1	3	3	3	4	4	3	31	77,5
8	4	4	3	0	3	4	4	4	3	2	31	77,5
9	3	4	4	1	3	3	3	4	3	3	31	77,5
10	3	3	3	2	3	2	3	3	3	3	28	70
11	4	3	3	0	4	4	4	3	3	4	32	80
12	3	3	4	1	4	4	4	4	4	3	33	82,5
13	4	4	3	1	3	4	3	3	3	4	32	80
14	4	4	4	1	3	3	4	4	4	3	34	85
15	3	4	3	0	3	3	4	4	4	3	31	77,5
16	4	4	3	3	3	3	2	2	3	2	29	72,5
17	4	4	3	1	3	4	3	3	3	4	32	80
18	4	4	4	1	4	3	4	3	3	3	33	82,5

Table 3 merupakan hasil perhitungan nilai *SUS* pada masing-masing responden, untuk nilai *SUS* paling rendah dari 18 responden ialah 72,5 yang menurut user tersebut sistem

Acceptable atau mendapat *Adjective Ratings Good*. Sedangkan secara keseluruhan sistem mendapatkan rata-rata nilai dari 18 responden ialah 79,58.

4. PENUTUP

Berdasarkan hasil penelitian dan rancang bangun sistem informasi manajemen magang berbasis *website*, dapat membantu meningkatkan efektifitas pelaksanaan program magang di Odama Studio selain itu membantu meningkatkan produktivitas user. Penelitian tersebut menggunakan dua pengujian sistem yaitu pengujian *Black Box* dan pengujian *SUS (System Usability Scale)*. Hasil pengujian *Black Box* yang diperoleh sesuai dengan rencana awal dan hasil pengujian *SUS* memperoleh nilai rata-rata 79,58 yang dapat dikatakan *Acceptable* dan memperoleh nilai *Grade Scale C*, dengan *Adjective Ratings Good* (Brooke, 2013). Hasil tersebut menunjukkan bahwa sistem layak digunakan atau sistem bisa diterima oleh user.

DAFTAR PUSTAKA

- Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. *IJCSI International Journal of Computer Science Issues*, 12(1), 106–111. https://www.academia.edu/10793943/A_Comparison_Between_Three_SDLC_Models_Waterfall_Model_Spiral_Model_and_Incremental_Iterative_Model
- Brooke, J. (1995). SUS: A “Quick and Dirty” Usability Scale. *Usability Evaluation In Industry*, July, 207–212. <https://doi.org/10.1201/9781498710411-35>
- Brooke, J. (2013). SUS: A Retrospective. *Journal of Usability Studies*, 8(2), 29–40.
- Gunanto, A., & Sudarmilah, E. (2020). Pengembangan Website E-Arsip di Kantor Kelurahan Pabelan. *Emitor: Jurnal Teknik Elektro*, 20(02), 104–110. <https://doi.org/10.23917/emitor.v20i02.10976>
- Kaban, E., Brata, K. C., & Brata, A. H. (2019). Evaluasi Usability Menggunakan Metode System Usability Scale (SUS) Dan Discovery Prototyping Pada Aplikasi PLN Mobile. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIHK) Universitas Brawijaya*, 3(2), 8952–8958.
- Pratama, I. G. N. W., Putra, I. G. . A. C., & Datya, A. I. (2017). Sistem Informasi Manajemen Praktek Kerja Lapangan Berbasis Website (Studi Kasus Program Studi Sistem Informasi Universitas Dhyana Pura Bali) I Gede Ngurah Wira Pratama 1) I.G.N Anom Cahyadi Putra 2) Aulia Iefan Datya 3). *Jurnal Teknologi Informasi Dan Komputer*, 3(1), 342–351.
- Ratnasari, L., & Lieli, S. (2017). *Sistim Pengelolaan Program Magang Bagi Mahasiswa Di Perusahaan (Studi Terhadap Tiga Perusahaan Di Jawa Tengah)*. 18–19.
- Royce, W. W. (1970). Managing the Development of Large Software Systems (1970). *Ideas That Created the Future*, August, 328–338.

<https://doi.org/10.7551/mitpress/12274.003.0035>

Samidjo, S. (2017). Efektifitas Pelaksanaan Magang Industri Mahasiswa Program Studi Pendidikan Teknik Mesin. *Taman Vokasi*, 2(2), 246.
<https://doi.org/10.30738/jtvok.v5i2.2528>

Suharyanti, C., Murtini, W., & Susilowati, T. (2015). *Pengaruh Proses Pembelajaran dan Program Kerja Praktek Terhadap Pengembangan Soft Skills Mahasiswa*. 4(7).

Supriyono, S. (2020). Software Testing with the approach of Blackbox Testing on the Academic Information System. *International Journal of Information System and Technology*, 3(2), 227–233.