

PENGEMBANGAN APLIKASI MOBILE GLORI-GROUP DENGAN EVALUASI USABILITY MENGGUNAKAN SYSTEM USABILITY SCALE

Charles Bintang*, Supardianto**

* Teknologi Rekayasa Multimedia, Politeknik Negeri Batam

** Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Batam

Article Info

Article history:

Received Jun 12th, 201x

Revised Aug 20th, 201x

Accepted Aug 26th, 201x

Keyword:

Mobile application

Android

CRM

Scrum

Usability testing

ABSTRACT

This research aims to develop a mobile-based application named "Glori-group" for PT Glori Global Group to address the limitations of the existing web-based CRM system. The web version lacked mobility support, and was only accessible via desktop browsers, which limited managers ability to monitor job statuses on the go. To solve this problem, an Android-based mobile application was designed using the Scrum development methodology to ensure iterative progress aligned with priority and needs. The application included features such as user management, customer tracking, campaign reporting, and lead documentation. The usability of the mobile application was evaluated using the System Usability Scale (SUS) to ensure that the new system improved accessibility and user experience. The results indicated that the mobile version effectively enhanced job monitoring efficiency, mobility, and communication within the organization

Corresponding Author:

Supardianto,

Informatics Engineering,

Batam State Polytechnic,

Ahmad Yani St, Teluk Tering, Kec Batam Kota, Batam, Riau Island, 29461, Indonesia

Email: supardianto@polibatam.ac.id

1. PENDAHULUAN

Perusahaan yang bergerak di bidang pemasaran produk impor seperti PT Glori Global Group sangat bergantung pada tingkat produktivitas dan efektivitas komunikasi antar tim, khususnya dalam hal interaksi dengan pelanggan. Salah satu alat bantu yang telah digunakan perusahaan ini adalah aplikasi *Customer Relationship Management (CRM)* berbasis *website* yang bernama "Glori-group". Aplikasi ini dirancang untuk mencatat aktivitas karyawan *marketing* serta mendukung manajer *marketing* dalam memantau status pekerjaan karyawan. Namun, pengembangan dan penggunaan aplikasi ini masih terbatas pada perangkat *desktop* dan *browser*, sehingga kurang mendukung mobilitas para manajer di lapangan, khususnya ketika dalam perjalanan bisnis.

Keterbatasan ini menyebabkan kebutuhan akan pengembangan aplikasi *mobile* yang mampu memberikan aksesibilitas lebih tinggi. Hasil pengumpulan data menunjukkan bahwa para manajer membutuhkan antarmuka *mobile* yang dapat diakses secara langsung melalui aplikasi *Android*, bukan melalui *browser*. Aplikasi versi *mobile* diharapkan dapat meningkatkan efisiensi dalam memantau aktivitas karyawan kapan pun dan di mana pun.

Sistem yang dikembangkan adalah aplikasi *mobile* berbasis *Android* yang berfungsi sebagai antarmuka *client-side* dari sistem *Customer Relationship Management (CRM)* 'Glori-group' berbasis *website* per desember 2024, dan tanpa modifikasi pada *backend* yang telah ada. Aplikasi ini ditujukan untuk manajer *marketing* agar dapat memantau aktivitas karyawan secara lebih fleksibel di luar kantor. Sistem ini menyediakan fitur seperti login, monitoring aktivitas harian, melihat dan mencatat tugas karyawan, serta menampilkan catatan yang dibuat oleh manajer maupun karyawan.

Pengembangan aplikasi ini dirancang menggunakan platform *Android Studio* dan mengikuti metodologi *Scrum* sebagai bagian dari kerangka *Agile*. *Scrum* dipilih untuk menyesuaikan dengan kebutuhan pengguna dan kemampuannya dalam membagi proses pengembangan menjadi iterasi-iterasi kecil yang terorganisir berdasarkan prioritas fitur dan kebutuhan pengguna [1]. Lalu untuk menjamin kualitas dari sisi pengalaman pengguna (*user experience*), aplikasi akan diuji menggunakan metode *System Usability Scale (SUS)*, yang merupakan salah satu instrumen standar dalam mengevaluasi aspek *usability* sebuah sistem [2].

Penelitian ini berfokus pada pengembangan aplikasi *mobile* "Glori-group" berbasis *Android* sebagai antarmuka tambahan bagi manajer *marketing*, yang tetap terhubung dengan *backend* dari sistem *CRM* berbasis *website* yang telah ada. Tujuan utama dari pengembangan ini adalah untuk meningkatkan aksesibilitas dan mobilitas manajer *marketing* dalam memantau pekerjaan karyawan serta mengevaluasi tingkat *usability* dari aplikasi melalui pendekatan kuantitatif menggunakan *SUS*.

2. METODE PENELITIAN

Dalam perancangan dan pengembangan aplikasi *mobile* "Glori-group" berbasis *android* ini, peneliti menggunakan metode pengumpulan data dengan melakukan wawancara yang bertujuan untuk memahami kebutuhan pengguna, dan memvalidasi masalah. Lalu metode pengembangan aplikasi yang digunakan adalah *scrum (agile framework)*, dan metode uji untuk menganalisa *Usability* yang digunakan adalah model *System Usability Scale (SUS)*.

2.1. Metode Pengumpulan Data

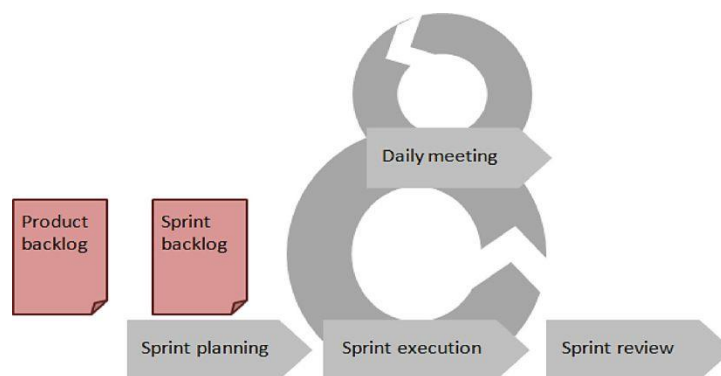
Peneliti menggunakan metode pengumpulan data dengan melakukan wawancara tujuannya untuk memvalidasi masalah yang terjadi, dan memahami kebutuhan pengguna untuk pengembangan aplikasi. Proses wawancara ini melibatkan interaksi langsung antara peneliti dan narasumber yaitu manajer *marketing*.

2.2. Metode Pengembangan Scrum

Pengembangan aplikasi *mobile* "Glori-group" berbasis *android* ini akan menggunakan metode pengembangan *scrum* yang merupakan pengembangan dari *framework Agile* dan mengedepankan kemampuan adaptasi yang cepat sesuai dengan kebutuhan pengguna [1]. Pada penerapan *Scrum* dibutuhkan *product backlog*, *sprint backlog*, *sprint planning*, *daily meeting*, *sprint execution* dan *sprint review*.

Product backlog merupakan pekerjaan-pekerjaan yang memiliki prioritas berdasarkan tingkat kebutuhan yang paling penting [2]. Dalam *product backlog* juga dilihat pada tingkat utama yang harus dikerjakan terlebih dahulu. Sehingga fitur utamalah yang menjadi acuan untuk pengerjaan di iterasi awal.

Sprint backlog merupakan tahap setelah pendefinisian *product backlog* yang selanjutnya akan dikembangkan dalam *sprint* [3]. Dapat disebut juga sebagai pengembangan produk pada awal inputan yang masuk. Pada *sprint backlog* dapat diketahui target atau persyaratan apa saja yang harus dipenuhi dalam pengembangan produk pada tiap iterasi. Pada Gambar 1 merupakan tahapan *scrum* dalam pengembangan produk.

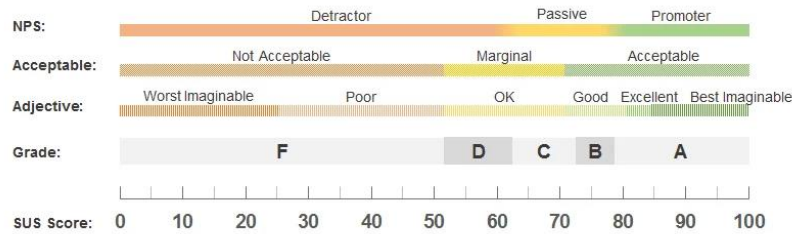


Gambar 1. Scrum proses [4]

2.3. Metode Uji Usability dengan menggunakan model System Usability Scale (SUS)

Setelah pengembangan aplikasi selesai, peneliti akan menilai *usability* dari aplikasi tersebut untuk memenuhi kebutuhan pengguna. Penilaian *usability* akan menggunakan *System Usability Scale (SUS)* yang implementasinya adalah distribusi kuesioner *SUS* kepada orang-orang yang telah menggunakan aplikasi termasuk manajer *marketing* di PT. Glori Global Group. *System Usability Scale (SUS)* adalah alat ukur yang digunakan untuk menilai *usability* terhadap sebuah produk, aplikasi atau sistem [5]. *SUS* terdiri dari sepuluh pertanyaan yang masing-masing pertanyaan memiliki skala lima poin yang berkisar dari "Sangat Tidak Setuju"

hingga “Sangat Setuju.” Terdapat lima pernyataan positif dan lima pernyataan negatif. Nilai *SUS* diinterpretasikan oleh *Jeff Sauro* dengan peringkat prosentase (*percentile ranks*) dan juga kelas huruf (*letter grades*) dari A sampai dengan F, dimana A adalah kelas terbaik dan F adalah kelas terburuk [6]. Namun, untuk penelitian ini, peneliti memilih kelas huruf (*letter grade*) saja. Gambar 2 menampilkan parameter yang digunakan untuk *SUS*.



Gambar 2. Skor *SUS* [7]

Ketentuan untuk letter grades adalah sebagai berikut:

1. *Grade A*: nilai ≥ 80.3
2. *Grade B*: $74 \leq \text{nilai} < 80.3$
3. *Grade C*: $68 \leq \text{nilai} < 74$
4. *Grade D*: $51 \leq \text{nilai} < 68$
5. *Grade F*: nilai < 51

Daftar pertanyaan yang digunakan dalam penelitian ini juga mengacu kepada metode pengujian *System Usability Scale (SUS)* yang terdiri dari 10 pertanyaan seperti pada Tabel 1.

Tabel 1. Daftar Pertanyaan Kuesioner [8]

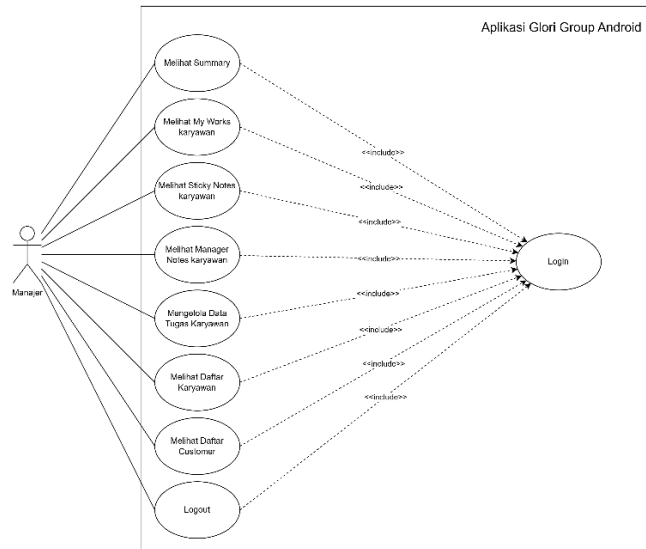
No	Pertanyaan
1	Saya berpikir akan menggunakan sistem ini lagi
2	Saya merasa sistem ini rumit untuk digunakan
3	Saya merasa sistem ini mudah digunakan
4	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini
5	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya
6	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

3. HASIL DAN ANALISIS

Pengembangan aplikasi dimulai ketika masalah telah divalidasi, dan kebutuhan pengguna telah didapatkan melalui wawancara. Pengembangan akan dimulai dengan pendefinisian *use case*, skenario *use case*, *entity relationship diagram*, lalu dilanjutkan dengan metode pengembangan *scrum*, dan analisis aplikasi menggunakan *System Usability Scale (SUS)*.

3.1. Perancangan sistem

Peneliti terlebih dahulu mendefinisikan kebutuhan pengguna dalam *use case diagram* seperti pada Gambar 3, dan menjelaskan masing-masing skenario *use case*.



Gambar 3. *Use case diagram*

Use case login bukan merupakan pusat dari semua *use case* dalam sistem, melainkan sebuah langkah autentikasi awal yang wajib dilakukan oleh manajer sebelum mendapatkan akses ke fitur-fitur utama lainnya. Oleh karena itu, posisinya pada *use case diagram* terhubung langsung ke beberapa fitur, namun tidak bermakna sebagai pusat utama dari sistem [9].

Skenario *use case login* adalah manajer (aktor) masuk ke dalam aplikasi menggunakan akun yang telah terdaftar sebagai manajer. Berikut prosedurnya.

1. Manajer membuka aplikasi mobile “Glori-group” berbasis *android*.
2. Sistem menampilkan form *login*.
3. Manajer mengisi *email* atau *username*, dan *password*.
4. Sistem melakukan verifikasi akun dengan role “Marketing Manager”.
5. Jika valid, manajer diarahkan ke halaman home.
6. Jika tidak valid, sistem menampilkan pesan kesalahan.

Skenario *use case* melihat *summary* adalah manajer (aktor) melihat data *summary* aktivitas hari tersebut seperti *new leads*, *new prospects*, *new customers*, *new campaigns*, dan *campaign activities*. Manajer juga dapat melihat aktivitas tiap karyawan seperti *leads*, *prospects*, *customers*, *campaigns*, dan *campaign activities*. Berikut prosedurnya.

1. Manajer berada dalam halaman *home*.
2. Sistem mengambil dan menampilkan data *summary*, dan data *summary* tiap karyawan berdasarkan tanggal hari tersebut.
3. Manajer dapat melihat *new leads*, *new prospects*, *new customers*, *new campaigns*, dan *campaign activity*.

Skenario *use case* melihat *my works* karyawan adalah manajer (aktor) melihat daftar tugas karyawan. Berikut prosedurnya.

1. Manajer berada dalam halaman *user detail*.
2. Manajer klik tombol ikon *more vertical*.
3. Sistem akan menampilkan 4 opsi, yaitu “Manager Notes”, “Note Log”, “Sticky Notes”, “My Works”.
4. Manajer klik teks “My Works”.
5. Sistem akan menampilkan *modal bottom sheet* yang berisi daftar tugas karyawan

Skenario *use case* melihat *sticky notes* karyawan adalah manajer (aktor) melihat catatan-catatan yang dibuat oleh karyawan terkait pekerjaan. Berikut prosedurnya:

1. Manajer berada dalam halaman *user detail*.
2. Manajer klik tombol ikon *more vertical*
3. Sistem akan menampilkan 4 opsi, yaitu “Manager Notes”, “Note Log”, “Sticky Notes”, “My Works”.
4. Manajer klik teks “Sticky Notes”.
5. Sistem akan menampilkan *modal bottom sheet* yang berisi catatan-catatan yang dibuat oleh karyawan terkait pekerjaan.

Skenario *use case* melihat *manajer notes* karyawan adalah manajer (aktor) melihat catatan-catatan yang manajer buat untuk masing-masing karyawan. Berikut prosedurnya:

1. Manajer berada dalam halaman *user detail*.
2. Manajer klik tombol ikon *more vertical*.
3. Sistem akan menampilkan 4 opsi, yaitu “Manager Notes”, “Note Log”, “Sticky Notes”, “My Works”.
4. Manajer klik teks “Manager Notes”.
5. Sistem akan menampilkan *modal bottom sheet* yang berisi catatan-catatan yang dibuat oleh manajer terkait pekerjaan.

Skenario *use case* mengelola data tugas karyawan adalah manajer (aktor) melakukan manajemen data tugas karyawan seperti menambah, mengedit, dan menghapus. Berikut prosedurnya:

1. Manajer berada dalam halaman *lead*.
2. Manajer klik salah satu card *lead*.
3. Sistem akan mengarahkan ke halaman *prospect list*. Lalu mengambil dan menampilkan data yang berkaitan dengan tugas tersebut.
4. Manajer dapat melakukan tindakan menambah gambar tugas atau *check note*, mengedit *check note* yang ada, dan menghapus tugas.
5. Sistem akan menyimpan perubahan yang dilakukan.

Skenario *use case* melihat daftar karyawan adalah manajer (aktor) melihat daftar karyawan dan dapat melakukan filter pada daftar tersebut. Berikut prosedurnya:

1. Manajer berada dalam halaman *user*.
2. Sistem mengambil dan menampilkan daftar karyawan.
3. Manajer dapat melakukan filter pada daftar tersebut.
4. Sistem mengambil dan menampilkan daftar karyawan yang cocok.

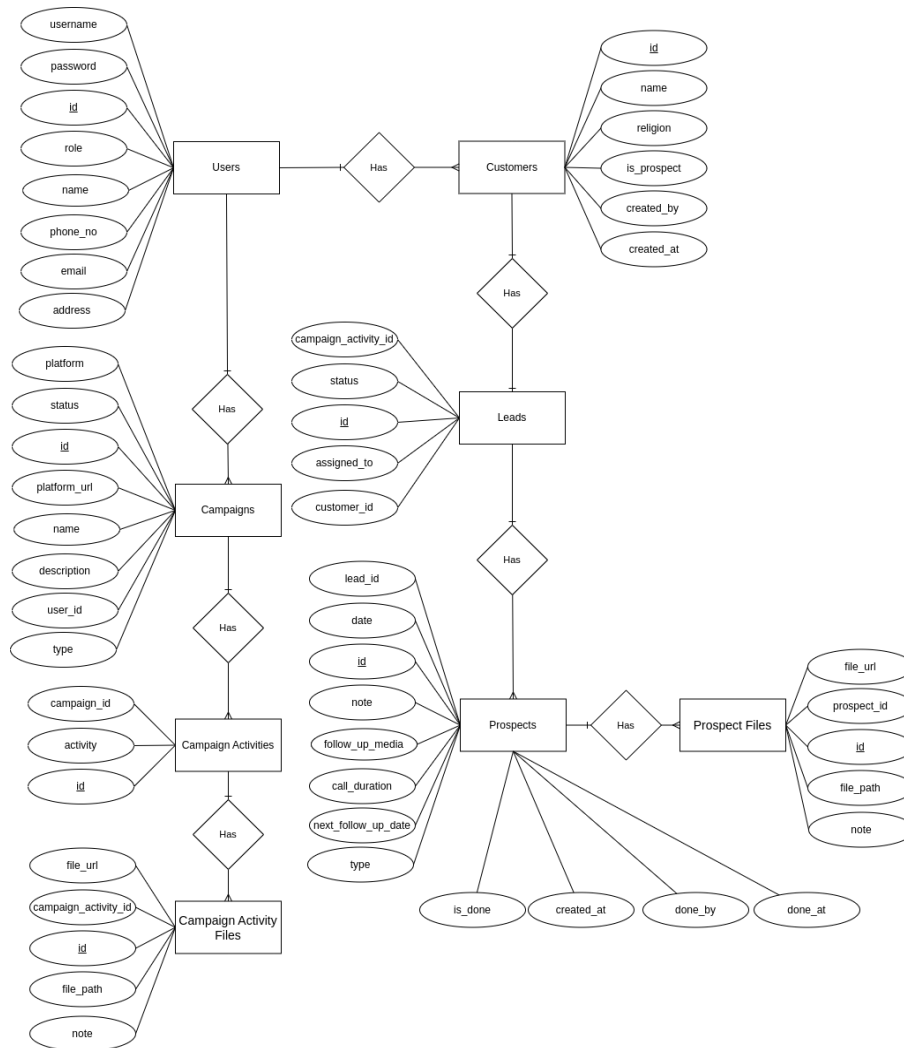
Skenario *use case* melihat daftar *customer* adalah manajer (aktor) melihat daftar *customer* dan dapat melakukan filter pada daftar tersebut. Berikut prosedurnya:

1. Manajer berada dalam halaman *user*.
2. Manajer klik teks “User List”.
3. Sistem akan menampilkan 2 opsi pilihan, yaitu “User List”, dan “Customer List”
4. Manajer klik opsi pilihan “Customer List”.
5. Sistem mengambil dan menampilkan daftar *customer*.
6. Manajer dapat melakukan filter pada daftar tersebut.
7. Sistem mengambil dan menampilkan daftar *customer* yang cocok.

Skenario *use case* logout adalah manajer (aktor) keluar dari aplikasi untuk mengakhiri sesi penggunaannya. Berikut prosedurnya:

1. Manajer klik *icon logout* dari halaman *profile*.
2. Sistem akan menampilkan dialog konfirmasi.
3. Manajer klik teks “Confirm”.
4. Sistem menghapus seluruh data sesi, dan mengarahkan manajer ke halaman *login*.

Diagram hubungan entitas adalah fondasi dari sebuah sistem yang menggambarkan struktur data dan hubungan antara entitas yang berbeda dalam basis data. Diagram hubungan entitas pada aplikasi ini telah didefinisikan seperti pada Gambar 4.



Gambar 4. Diagram Hubungan Entitas

Daftar lengkap *application programming interface (API)* yang digunakan untuk pengembangan aplikasi ini dapat dilihat di lampiran. Setiap *endpoint* mendukung fitur-fitur utama aplikasi seperti *login*, pengelolaan data karyawan, pencatatan aktivitas, dan pengelolaan *prospect* serta *customer*.

Perancangan antarmuka pengguna aplikasi *mobile* “Glori-group” didasarkan pada desain UI dari sistem “Glori-group” versi *website* yang telah digunakan oleh karyawan dan manajer marketing. Tujuan dari pendekatan ini adalah untuk menjaga konsistensi tampilan dan alur navigasi, sehingga pengguna yang sudah terbiasa dengan versi web dapat dengan cepat beradaptasi dengan versi *mobile* [10].

3.2. Implementasi metode scrum dalam pengembangan aplikasi

Implementasi *scrum* dimulai dengan mendefinisikan *product backlog* yang merupakan pengelompokan kebutuhan aplikasi yang akan dikembangkan. Berdasarkan *use case* pada Gambar 2, dan skenario *use case*, maka *product backlog* akan menyesuaikan dengan tingkatan fitur yang akan dikerjakan terlebih dahulu. Tingkatan ini dilihat dari urgensi dari kebutuhan user. Hasil dari tahapan *product backlog* yang berisi tentang fitur mana yang harus dikerjakan terlebih dahulu berdasarkan prioritasnya disajikan di Tabel 2.

Tabel 2. *Product Backlog*

No	Fitur	Prioritas
----	-------	-----------

1	Autentikasi <i>User</i>	<i>Low</i>
2	Melihat <i>sticky notes</i> karyawan	<i>Low</i>
3	Melihat <i>manager notes</i> karyawan	<i>Low</i>
4	Melihat <i>my works</i> karyawan	<i>Low</i>
5	Melihat daftar <i>customer</i>	<i>Medium</i>
6	Melihat daftar karyawan	<i>Medium</i>
7	Mengelola data tugas karyawan	<i>High</i>
8	Melihat <i>summary</i> karyawan	<i>High</i>
9	Perancangan tampilan aplikasi	<i>High</i>

Tahapan selanjutnya adalah *sprint* yang terdiri dari *sprint planning*, *sprint backlog*, *sprint execution*, dan *daily meeting*. Dalam tahapan *sprint* ini dikerjakan terlebih dahulu fitur yang menjadi prioritas utama atau *high*, kemudian dilanjutkan dengan tingkatan *medium* dan *low*. Tahapan *sprint planning* disini akan dijelaskan dan dijabarkan per fitur berdasarkan *sprint backlog* yang mendefinisikan target atau persyaratan (*Definition of Done*) apa saja yang harus dipenuhi yang akan dijalankan untuk menyelesaikan *sprint*. Berdasarkan *product backlog* maka berikut adalah hasil dari tahapan *sprint planning* dan *sprint backlog* yang akan dijalankan terdapat pada Tabel 3.

Tabel 3. *Sprint planning*

No	Fitur	<i>Sprint Backlog</i>	<i>Sprint Planning</i>
1	Perancangan tampilan aplikasi	<i>Definition of Done:</i> a. merancang tampilan aplikasi b. implementasi rancangan tampilan aplikasi	Sprint 1 bertujuan untuk menyelesaikan tampilan aplikasi dan implementasinya di <i>Android Studio</i> . Direncanakan berlangsung selama 4 minggu, hasil akhir diperkirakan adalah figma yang mencakup semua tampilan aplikasi, dan tahap awal aplikasi yang hanya berupa <i>User Interface</i> dari seluruh tampilan yang telah dirancang.
2	Melihat <i>summary</i> dari karyawan	<i>Definition of Done:</i> a. menampilkan <i>summary</i> dari <i>new leads</i> , <i>new prospects</i> , <i>new customers</i> , <i>new campaigns</i> , dan <i>campaign activities</i> , berdasarkan tanggal yang dipilih b. menampilkan <i>summary</i> dari setiap karyawan berdasarkan tanggal yang dipilih	Sprint 2 bertujuan untuk menyelesaikan fitur <i>summary</i> . Direncanakan berlangsung selama 1 minggu, hasil akhir diperkirakan adalah manajer dapat melihat <i>summary</i> dan <i>summary</i> tiap karyawan di aplikasi.
3	Mengelola data tugas karyawan	<i>Definition of Done:</i> a. menampilkan <i>campaigns</i> , <i>campaign activities</i> , <i>leads</i> , dan <i>prospect</i> b. menambahkan gambar pada data <i>prospect</i>	Sprint 3 bertujuan untuk menyelesaikan fitur <i>CRUD</i> pada <i>prospect</i> , menampilkan data <i>campaigns</i> , data <i>campaign activities</i> , dan data <i>leads</i> . Direncanakan berlangsung selama 3 minggu, hasil

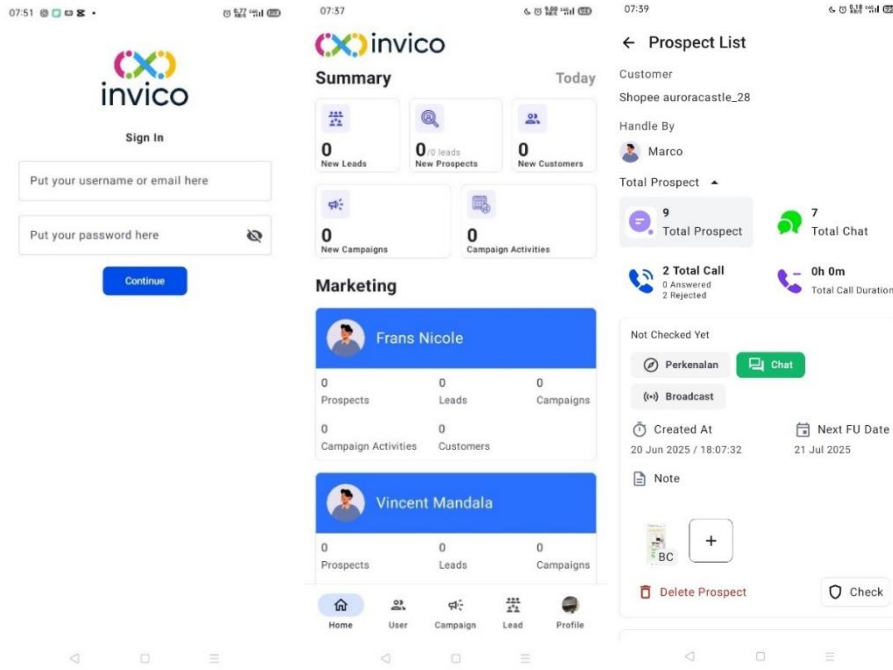
		<ul style="list-style-type: none"> c. menghapus data <i>prospect</i> d. menandai bahwa data <i>prospect</i> sudah di-<i>check</i> e. memberi catatan pada data <i>prospect</i> 	akhir diperkirakan adalah manajer dapat melihat tampilan <i>campaigns</i> , <i>campaign activities</i> , <i>leads</i> , <i>prospect</i> , memberi catatan (<i>manager note</i>), menambahkan gambar, dan menghapus data <i>prospect</i> di aplikasi.
4	Melihat daftar <i>customer</i>	<p><i>Definition of Done:</i></p> <ul style="list-style-type: none"> a. menampilkan daftar <i>customer</i> b. menampilkan daftar <i>customer</i> yang sudah di-<i>filter</i> berdasarkan nama 	Sprint 4 bertujuan untuk menyelesaikan fitur <i>customer</i> . Direncanakan berlangsung selama 1 minggu, hasil akhir diperkirakan adalah manajer dapat melihat tampilan <i>customer</i> , dan melakukan <i>filter</i> pada daftar <i>customer</i> tersebut.
5	Melihat daftar karyawan	<p><i>Definition of Done:</i></p> <ul style="list-style-type: none"> a. menampilkan daftar karyawan b. menampilkan daftar karyawan yang sudah di-<i>filter</i> berdasarkan nama 	Sprint 5 bertujuan untuk menyelesaikan fitur karyawan. Direncanakan berlangsung selama 1 minggu, hasil akhir diperkirakan adalah manajer dapat melihat tampilan karyawan, dan melakukan <i>filter</i> pada daftar karyawan tersebut.
6	Melihat <i>my works</i> karyawan	<p><i>Definition of Done:</i></p> <ul style="list-style-type: none"> a. menampilkan <i>my works</i> atau <i>todo-list</i> dari karyawan berdasarkan kategori keterangan waktu (<i>today</i>, <i>past</i>, <i>all</i>) 	Sprint 6 bertujuan untuk menyelesaikan fitur <i>my works</i> atau <i>todo-list</i> . Direncanakan berlangsung selama 1 minggu, hasil akhir diperkirakan adalah manajer dapat melihat tampilan <i>my works</i> , dan melakukan <i>filter</i> berdasarkan kategori keterangan waktu (<i>today</i> , <i>past</i> , <i>all</i>).
7	Melihat <i>manager notes</i> karyawan	<p><i>Definition of Done:</i></p> <ul style="list-style-type: none"> a. menampilkan catatan yang dibuat manajer kepada karyawan 	Sprint 7 bertujuan untuk menyelesaikan fitur <i>manager note</i> . Direncanakan berlangsung selama 1 minggu, hasil akhir diperkirakan adalah manajer dapat melihat <i>manager note</i> atau catatan yang dibuat manajer kepada karyawannya, dan ketika diklik akan mengarahkan manajer ke data <i>prospect</i> yang sesuai.
8	Melihat <i>sticky notes</i> karyawan	<p><i>Definition of Done:</i></p> <ul style="list-style-type: none"> a. menampilkan catatan yang dibuat karyawan 	Sprint 8 bertujuan untuk menyelesaikan fitur <i>sticky note</i> . Direncanakan berlangsung selama 1 minggu, hasil akhir diperkirakan adalah manajer dapat melihat <i>sticky note</i> atau catatan yang dibuat karyawannya terkait pekerjaan.
9	Autentikasi <i>User</i>	<p><i>Definition of Done:</i></p> <ul style="list-style-type: none"> a. hanya manajer yang dapat <i>login</i> dan <i>logout</i> 	Sprint 9 bertujuan untuk menyelesaikan fitur autentikasi. Direncanakan berlangsung selama 2 minggu, hasil akhir diperkirakan adalah manajer dapat <i>login</i> ke aplikasi dan <i>logout</i> dari aplikasi.

Kemudian dilakukan tahapan *sprint execution* yang merupakan tahapan pengembangan dan *daily meeting* sesuai dengan *sprint planning* dan *sprint backlog* tiap iterasinya. Pertanyaan-pertanyaan yang diajukan didalam *daily meeting* melacak kemajuan pengembangan [11]. Daftar catatan *daily meeting* dapat dilihat di lampiran. Tabel 4 menampilkan hasil dari *sprint execution*.

Tabel 4. *Sprint execution*

No	Fitur	Task	Status
1	Perancangan tampilan aplikasi	<i>Definition of Done:</i> a. merancang tampilan aplikasi b. implementasi rancangan tampilan aplikasi	<i>Done</i>
2	Melihat summary dari karyawan	<i>Definition of Done:</i> a. menampilkan <i>summary</i> dari <i>new leads, new prospects, new customers, new campaigns, dan campaign activities</i> , berdasarkan tanggal yang dipilih b. menampilkan <i>summary</i> dari setiap karyawan berdasarkan tanggal yang dipilih	<i>Done</i>
3	Mengelola data tugas karyawan	<i>Definition of Done:</i> a. menampilkan <i>campaigns, campaign activities, leads, dan prospect</i> b. menambahkan gambar pada data <i>prospect</i> c. menghapus data <i>prospect</i> d. menandai bahwa data <i>prospect</i> sudah di-check e. memberi catatan pada data <i>prospect</i>	<i>Done</i>
4	Melihat daftar <i>customer</i>	<i>Definition of Done:</i> a. menampilkan daftar <i>customer</i> b. menampilkan daftar <i>customer</i> yang sudah di-filter berdasarkan nama	<i>Done</i>
5	Melihat daftar karyawan	<i>Definition of Done:</i> a. menampilkan daftar karyawan b. menampilkan daftar karyawan yang sudah di-filter berdasarkan nama	<i>Done</i>
6	Melihat <i>my works</i> karyawan	<i>Definition of Done:</i> a. menampilkan <i>my works</i> atau <i>todo-list</i> dari karyawan berdasarkan kategori keterangan waktu (<i>today, past, all</i>)	<i>Done</i>
7	Melihat <i>manager notes</i> karyawan	<i>Definition of Done:</i> a. menampilkan catatan yang dibuat manajer kepada karyawan	<i>Done</i>
8	Melihat <i>sticky notes</i> karyawan	<i>Definition of Done:</i> a. menampilkan catatan yang dibuat karyawan	<i>Done</i>
9	Autentikasi <i>User</i>	<i>Definition of Done:</i> a. hanya manajer yang dapat <i>login</i> dan <i>logout</i>	<i>Done</i>

Tahapan *sprint review* adalah tahapan setelah *sprint execution* tiap iterasi *sprint*. Dalam tahapan ini akan dilakukan demonstrasi fitur kepada manajer, dan masukan dari manajer akan dicatat untuk pengembangan iterasi selanjutnya. Berikut adalah hasil dari keseluruhan *sprint review* yang disajikan dalam gambar *user interface* aplikasi dari fitur-fitur yang sudah memenuhi *definition of done*.



Gambar 5. Halaman login, home, dan prospect list

Gambar 5 merupakan halaman login yang pertama yang ditampilkan ketika manajer belum pernah login ke aplikasi. Manajer bisa input username dan password akun, dan tekan tombol continue untuk verifikasi akun. Halaman home ditampilkan ketika manajer berhasil masuk ke dalam aplikasi yang berisi summary dan summary tiap karyawan. Halaman prospect list merupakan halaman detail dari lead yang menampilkan setiap prospect dari customer.

3.3. Analisis Usability dengan menggunakan model System Usability Scale (SUS)

Setelah pengembangan aplikasi selesai, peneliti melibatkan 31 responden untuk mengumpulkan data berupa kuesioner yang akan diolah sesuai dengan SUS [5]. 31 responden tersebut terdiri dari 4 manajer marketing dan 27 karyawan marketing yang sudah terbiasa dengan antarmuka website “Glori-group” sebagai pengguna awal. Tujuan melibatkan pengguna non-manajer adalah untuk memperoleh gambaran yang lebih luas mengenai usability dari sudut pandang pengguna baru dan menilai tingkat intuitivitas antarmuka aplikasi [12]. Data kuesioner akan diolah berdasarkan langkah-langkah yang sesuai dengan SUS [5] seperti berikut:

1. Skor tiap pertanyaan dihitung ulang terlebih dahulu. Untuk pertanyaan positif (ganjil) skor jawaban akan dikurangi 1, dan untuk pertanyaan negatif (genap) skor jawaban adalah hasil dari 5 dikurangi skor jawaban. Sehingga nilai setiap pertanyaan akan berada dalam rentang 0 sampai dengan 4.
2. Jumlah maksimal skor dari 10 pertanyaan adalah 40. Sehingga setiap jumlah skor dari responden akan dikalikan dengan 2,5 dan menghasilkan skor akhir atau nilai dalam rentang 0 sampai dengan 100.

Hasil pengolahan data kuesioner tersebut disajikan pada Tabel 6 berikut.

Tabel 5. Hasil Pengolahan Data

Email	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai
kho***@gmail.com	2	2	4	2	4	0	4	1	3	1	23	57,5
ini***@gmail.com	1	3	4	3	4	3	3	3	4	2	30	75
...
sho***@gmail.com	3	2	2	3	2	3	1	3	3	0	22	55

HASIL AKHIR	65,403
--------------------	--------

Setelah dilakukan pengolahan data untuk menilai *usability* aplikasi dari tiap-tiap responden, diperoleh hasil akhir sebesar 65.403 (*Grade D*) yang menunjukkan bahwa aplikasi mobile “Glori-group” berbasis android memiliki angka *usability* yang masih kurang baik. Diketahui bahwa skor 0, skor 1, dan skor 2 paling banyak berada di *Q10* yang menunjukkan bahwa pengguna baru memerlukan waktu lebih untuk mengenal dan menggunakan aplikasi. Kemudian skor 3 paling banyak berada di *Q5*, skor 4 paling banyak berada di *Q3* yang menunjukkan bahwa fitur-fitur aplikasi ini sudah berjalan dengan baik dan cukup mudah digunakan bagi sebagian responden [6].

4. KESIMPULAN

Berdasarkan hasil pengembangan aplikasi *mobile* “Glori-group” berbasis *android* yang sudah memenuhi *Definition of Done* dapat diperoleh kesimpulan bahwa aplikasi sudah telah sesuai dengan kebutuhan user yang meningkatkan aksesibilitas, dan mobilitas manajer *marketing* ketika melakukan perjalanan bisnis atau sedang berada diluar kantor dan perlu untuk memonitor pekerjaan karyawan *marketing*.

Kemudian berdasarkan pengujian *usability* dengan menggunakan model *SUS*, diketahui bahwa aplikasi ini memiliki nilai *usability* sebesar 65 atau *grade D* yang menunjukkan bahwa aplikasi ini sudah cukup *usable*, namun belum tergolong unggul, dan masih perlu penyempurnaan (*marginally acceptable*) [7].

Secara keseluruhan aplikasi *mobile* “Glori-group” berbasis *android* ini memberikan solusi dalam peningkatan aksesibilitas, dan mobilitas bagi manajer *marketing*. Namun masih kurang baik dalam hal *usability* dan memerlukan penelitian lebih lanjut untuk mencapai hasil *usability* yang lebih baik, sehingga aplikasi mudah diterima, dan tidak membingungkan pengguna baru.

REFERENCES

- [1] N. Rafianto, Dimas, and Saifulloh, “Penerapan Metode Scrum Pada Pembuatan User Experience Landing Page Sistem Informasi Lentera,” *Jurnal Sistem Informasi dan Sains Teknologi*, vol. 3, no. 2, pp. 1–14, Aug. 2021, doi: 10.31326/sistek.v3i2.979.
- [2] I. Kurniawan and R. R. Sani, “Pemodelan SCRUM dalam Pengembangan Sistem Informasi Kesehatan pada Klinik Ar-Rokhim Sragen Kabupaten Sragen SCRUM Model on Development of Health Information System at Ar-Rokhim Clinic in Sragen Regency,” *Journal of Information System*, vol. 4, no. 1, pp. 76–86, 2019.
- [3] M. Hron and N. Obwegeser, “Scrum in practice: an overview of Scrum adaptations,” in *Hawaii International Conference on System Sciences*, Curran Associates, Inc., 2018. [Online]. Available: <http://hdl.handle.net/10125/50568>
- [4] T. Agustina Nugrahani and K. Nine Amalia, “Implementasi Scrum dalam Perancangan Aplikasi Pembelajaran Budaya Nusantara berbasis Mobile,” 2022.
- [5] J. Brooke, “SUS: A quick and dirty usability scale,” Nov. 1995. [Online]. Available: <https://www.researchgate.net/publication/228593520>
- [6] J. Sauro, “Measuring Usability with the System Usability Scale (SUS).” Accessed: Nov. 28, 2024. [Online]. Available: <https://measuringu.com/sus/>
- [7] J. Sauro, “5 Ways to Interpret a SUS Score.” Accessed: Jun. 20, 2025. [Online]. Available: <https://measuringu.com/interpret-sus-score/>
- [8] M. Alvian Kosim, S. Restu Aji, and M. Darwis, “PENGUJIAN USABILITY APLIKASI PEDULILINDUNGI DENGAN METODE SYSTEM USABILITY SCALE (SUS) 1),” *Jurnal Sistem Informasi dan Sains Teknologi*, vol. 4, no. 2, Aug. 2022.
- [9] R. Juliarto, “Contoh Use Case Diagram Lengkap dengan Penjelasan.” Accessed: Jul. 23, 2025. [Online]. Available: <https://www.dicoding.com/blog/contoh-use-case-diagram/>
- [10] J. Nielsen, “Enhancing the Explanatory Power of Usability Heuristics,” Apr. 1994.
- [11] M. Paasivaara, S. Durasiewicz, and C. Lassenius, “Distributed agile development: Using Scrum in a large project,” in *Proceedings - 2008 3rd IEEE International Conference Global Software Engineering, ICGSE 2008*, 2008, pp. 87–95. doi: 10.1109/ICGSE.2008.38.
- [12] C. D. Sara and N. R. Nurwulan, “Comparative Usability Evaluation of Novice and Expert Gojek Users,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Sep. 2021, pp. 16–22. doi: 10.1145/3479645.3479650.