

BAB II

KAJIAN LITERATUR

2.1 Palang Merah Indonesia (PMI)

Palang Merah Indonesia (PMI) merupakan organisasi kemanusiaan nasional yang bergerak di bidang sosial dan kesehatan, khususnya dalam pengelolaan pelayanan darah. PMI berdiri pada 17 September 1945 dan menjadi bagian dari Gerakan Palang Merah dan Bulan Sabit Merah Internasional. Sebagai bagian dari organisasi kemanusiaan dunia, PMI menjalankan tugas berdasarkan Tujuh Prinsip Dasar Palang Merah Internasional, yaitu kemanusiaan, kesamaan, kenetralan, kemandirian, kesukarelaan, kesatuan, dan kesemestaan. Prinsip-prinsip ini menjadi landasan moral dalam seluruh pelayanan PMI, terutama dalam upaya penyelamatan nyawa melalui ketersediaan darah bagi masyarakat. Peran PMI dalam pelayanan darah diatur secara resmi dalam Peraturan Pemerintah Republik Indonesia Nomor 7 Tahun 2011 tentang Pelayanan Darah, yang menetapkan PMI sebagai satu-satunya lembaga yang berwenang dalam pengelolaan darah di Indonesia. Tugas tersebut mencakup kegiatan pengumpulan darah, pemeriksaan, pemrosesan, penyimpanan, distribusi, hingga pelayanan transfusi darah yang aman dan berkualitas.

Selain itu, PMI juga bertanggung jawab dalam meningkatkan pengetahuan, kesadaran, dan partisipasi masyarakat untuk mendonorkan darah secara sukarela dan rutin tanpa pamrih. Salah satu kegiatan utama PMI dalam pemenuhan stok darah nasional adalah penyelenggaraan donor darah sukarela, baik melalui Unit Transfusi Darah (UTD) maupun kegiatan donor darah keliling (*mobile unit*). Kegiatan ini dilaksanakan di berbagai lokasi seperti sekolah, universitas, instansi pemerintah, perusahaan, tempat ibadah, hingga fasilitas publik lainnya. Melalui kegiatan ini, PMI tidak hanya mengumpulkan darah, tetapi juga menjalankan fungsi edukasi dan sosialisasi mengenai pentingnya donor darah, manfaat kesehatan bagi pendonor, serta penerapan sistem transfusi yang aman dan beretika. Upaya ini menjadi aspek penting dalam membangun budaya donor darah secara berkelanjutan di masyarakat. Namun, pelaksanaan pelayanan donor darah masih menghadapi sejumlah tantangan, seperti rendahnya partisipasi pendonor rutin, kurangnya penyebaran informasi yang efektif, serta sistem pendataan pendonor yang masih dipengaruhi oleh metode manual di beberapa daerah. Kondisi ini dapat memengaruhi ketersediaan stok darah, terutama pada wilayah dengan keterbatasan sarana informasi. Pemanfaatan teknologi informasi menjadi

salah satu solusi strategis untuk meningkatkan efektivitas pelayanan PMI, khususnya dalam penyebaran informasi, pengelolaan data pendonor, dan pengingat masa donor berikutnya.

Penerapan sistem digital, seperti aplikasi donor darah berbasis Android dan sistem informasi berbasis web, dinilai mampu meningkatkan kualitas pelayanan PMI melalui pengelolaan data yang lebih terintegrasi, penyampaian informasi yang cepat dan tepat sasaran, serta penggunaan fitur push notification sebagai pengingat otomatis jadwal donor. Hal ini didukung oleh berbagai penelitian terdahulu, di antaranya penelitian di PMI Batam yang mengembangkan aplikasi stok darah berbasis Android untuk memudahkan masyarakat memperoleh informasi ketersediaan darah dan melakukan pemesanan [7], penelitian di UTD PMI Kabupaten Lumajang yang mengembangkan sistem informasi donor darah berbasis web untuk mengatasi keterbatasan akses data pendonor dan stok darah [8], serta penelitian di PMI Kota Palembang yang menawarkan solusi pendaftaran donor darah secara online guna mengurangi antrean dan meningkatkan minat donor rutin [9]. Selain itu, penelitian Nahak dkk. di Rumah Sakit Umum Kefamenanu yang bekerja sama dengan PMI juga menunjukkan bahwa digitalisasi sistem donor darah berbasis Android mampu meningkatkan efektivitas penyebaran informasi, pencatatan data pendonor, dan pengelolaan stok darah secara lebih terstruktur dan terdigitalisasi [10].

2.2 Sistem Informasi

Sistem informasi merupakan suatu kesatuan yang terdiri dari komponen manusia, perangkat keras, perangkat lunak, prosedur, dan data yang saling berinteraksi untuk mengumpulkan, mengolah, menyimpan, serta mendistribusikan informasi guna mendukung proses pengambilan keputusan dalam suatu organisasi. Sistem informasi bertujuan untuk menyediakan data yang tepat, akurat, dan relevan sehingga pengguna dapat mengakses informasi sesuai kebutuhan dalam waktu yang cepat. Dalam konteks pengembangan layanan publik, sistem informasi menjadi salah satu elemen penting untuk meningkatkan kualitas pelayanan melalui pemanfaatan teknologi digital.

Secara konseptual, sistem informasi merupakan gabungan dari konsep sistem dan informasi. Sistem dipahami sebagai sekumpulan elemen yang saling terhubung dan bekerja sama untuk mencapai tujuan tertentu, sedangkan informasi adalah data yang telah diolah sehingga memiliki makna dan nilai guna. Dengan demikian, sistem informasi tidak hanya berfungsi sebagai alat pengolahan data, tetapi juga sebagai sarana penyedia informasi yang mendukung efisiensi operasional, komunikasi, pengelolaan data, serta pengambilan keputusan dalam suatu organisasi. Sistem informasi berperan penting dalam membantu organisasi meningkatkan efisiensi operasional, komunikasi, pengelolaan data, serta

pelayanan kepada pengguna. Manfaat sistem informasi dapat dilihat dari beberapa aspek, antara lain:

1. Efisiensi Pengelolaan Data, di mana penyimpanan dan pemrosesan data dapat dilakukan secara otomatis dan terstruktur.
2. Kecepatan Akses Informasi, karena data dapat diakses kapan saja dan dari berbagai tempat melalui media digital.
3. *Error Minimization* (Pengurangan Kesalahan), karena data diproses oleh sistem sehingga mengurangi kesalahan manual.
4. Dukungan Pengambilan Keputusan, dengan menyediakan informasi yang akurat dan terkini.

Dalam bidang kesehatan dan pelayanan sosial, sistem informasi banyak dimanfaatkan sebagai sarana pendukung kegiatan administrasi, edukasi, dan monitoring karena mampu mengelola data pasien, stok darah, catatan medis, serta jadwal pelayanan secara lebih efektif dibandingkan metode manual. Seiring perkembangan teknologi mobile, sistem informasi tidak hanya diimplementasikan pada komputer, tetapi juga pada perangkat smartphone sehingga layanan publik, termasuk donor darah, dapat diakses secara lebih luas dan interaktif melalui aplikasi mobile. Dalam penelitian ini, sistem informasi diterapkan untuk mendukung kegiatan donor darah di PMI melalui pengembangan aplikasi berbasis Android yang memungkinkan pendonor memperoleh informasi jadwal donor, melihat riwayat pendonoran, serta menerima pengingat donor secara otomatis melalui fitur *push notification*, sehingga menjadi solusi strategis untuk meningkatkan efektivitas pelayanan donor darah dan mendorong partisipasi masyarakat secara berkelanjutan.

Berbagai penelitian terdahulu menunjukkan bahwa penerapan sistem informasi donor darah berbasis web dan Android memberikan dampak positif terhadap peningkatan kualitas pelayanan. Khairatunnisa dan Sari menyatakan bahwa sistem informasi donor darah online memudahkan masyarakat dalam mengakses informasi donor, melakukan pendaftaran dan penjadwalan secara daring, serta memantau stok darah secara real-time, sehingga meningkatkan partisipasi masyarakat dan efisiensi pelayanan transfusi darah [11]. Mukhlis juga menegaskan bahwa sistem informasi donor darah berbasis web mampu meningkatkan kecepatan akses informasi stok darah, memudahkan pencarian data pendonor, serta meningkatkan transparansi pelayanan PMI [8]. Selain itu, pengembangan sistem informasi donor darah dalam program pengabdian masyarakat di Cikarang Kota menunjukkan bahwa sistem tersebut mampu menyediakan informasi yang akurat dan terbaru, mendorong

partisipasi masyarakat, serta meningkatkan kesadaran akan pentingnya donor darah melalui penyajian informasi yang mudah diakses [12].

2.3 Aplikasi *Mobile*

Aplikasi mobile merupakan perangkat lunak yang dirancang untuk dijalankan pada perangkat bergerak seperti smartphone dan tablet. Aplikasi ini dikembangkan khusus untuk memanfaatkan kemampuan perangkat mobile, seperti konektivitas internet, sensor, kamera, GPS, serta sistem notifikasi yang tidak dimiliki oleh perangkat komputer tradisional. Dengan perkembangan teknologi digital dan penetrasi penggunaan smartphone yang semakin tinggi, aplikasi mobile kini menjadi salah satu media utama dalam penyampaian informasi, layanan publik, komunikasi, hingga aktivitas sosial masyarakat. Aplikasi mobile secara umum dapat dibedakan menjadi beberapa kategori berdasarkan fungsinya, seperti aplikasi produktivitas, hiburan, pendidikan, kesehatan, dan layanan sosial. Peran aplikasi mobile tidak hanya sebatas menyajikan informasi, tetapi juga menyediakan layanan interaktif yang memungkinkan pengguna melakukan aktivitas tertentu secara mandiri dan real-time. Hal ini menjadikan aplikasi mobile sebagai solusi yang efektif untuk meningkatkan efisiensi layanan, memperluas akses informasi, dan mempermudah interaksi antara pengguna dan penyedia layanan [13].

Kelebihan utama aplikasi mobile dibandingkan aplikasi berbasis web tradisional terletak pada sifatnya yang lebih personal, responsif, dan selalu terhubung dengan pengguna, sehingga memungkinkan penyampaian informasi secara langsung melalui fitur seperti *push notification* yang mendukung interaksi cepat dan tepat waktu. Karakteristik ini menjadikan aplikasi mobile memiliki potensi besar dalam meningkatkan keterlibatan pengguna (*user engagement*), terutama pada layanan yang membutuhkan pengingat berkala, penyampaian informasi penting, serta komunikasi langsung antara penyedia layanan dan pengguna. Dalam konteks layanan publik dan kesehatan, pemanfaatan aplikasi mobile dikenal sebagai *mobile health* (mHealth), yaitu penggunaan perangkat mobile untuk meningkatkan kualitas pelayanan kesehatan melalui manajemen data, edukasi, monitoring, serta penyebaran informasi yang efisien, sehingga memudahkan penyedia layanan dalam pengelolaan sistem sekaligus membantu masyarakat memperoleh layanan dan informasi kesehatan secara cepat dan mudah melalui perangkat yang digunakan sehari-hari.

Penggunaan aplikasi mobile turut memberikan kontribusi signifikan dalam meningkatkan efektivitas komunikasi antara lembaga penyedia layanan dan masyarakat, karena informasi dapat diakses dengan mudah serta disampaikan secara tepat waktu sehingga mendorong partisipasi masyarakat yang lebih tinggi terhadap layanan yang

ditawarkan. Dalam penelitian ini, pemanfaatan aplikasi mobile berbasis Android digunakan untuk mendukung layanan donor darah yang dikelola oleh PMI, di mana pendonor dapat memperoleh informasi mengenai jadwal dan lokasi kegiatan donor darah serta riwayat pendonoran secara lebih efisien. Selain itu, keberadaan fitur pengingat otomatis melalui notifikasi berperan sebagai dorongan bagi masyarakat untuk melakukan donor darah secara rutin. Secara keseluruhan, aplikasi mobile tidak hanya berfungsi sebagai media penyedia informasi, tetapi juga sebagai sarana untuk meningkatkan interaksi, partisipasi, dan kualitas pelayanan publik, sehingga pemanfaatannya dalam layanan donor darah menjadi strategi yang relevan dalam mengatasi permasalahan penyebaran informasi dan pendataan donor yang masih dilakukan secara manual serta membantu meningkatkan ketersediaan stok darah melalui partisipasi masyarakat yang lebih aktif dan terorganisir.

Pemanfaatan aplikasi mobile dalam konteks donor darah telah banyak dikaji dalam berbagai penelitian. Aplikasi stok darah PMI berbasis Android di Batam dirancang untuk menampilkan informasi ketersediaan darah, memfasilitasi pemesanan, serta menampilkan jumlah pendonor sehingga masyarakat tidak perlu datang langsung ke PMI hanya untuk mengecek stok darah [7]. Selain itu, aplikasi Donor Darah Online berbasis Android dan web memfasilitasi komunikasi antara pasien, rumah sakit, dan pendonor melalui sistem notifikasi berbasis lokasi guna mempercepat pencarian pendonor terdekat [11]. Pada kasus khusus, Irvan dan Fajri mengembangkan aplikasi donor plasma konvalesen berbasis Android dengan integrasi geofencing, Google Maps SDK, Firebase, dan kode QR untuk meningkatkan akurasi data, validasi pendonor, serta koordinasi antara pendonor dan fasilitas kesehatan [14]. Sementara itu, aplikasi Donora menyediakan platform donor darah darurat dengan fitur pelacakan *real-time*, sistem poin, dan gamifikasi untuk meningkatkan motivasi serta konsistensi pendonor dalam melakukan donor darah secara rutin [15].

2.4 Fitur Push-Notification

Push-notification merupakan salah satu fitur komunikasi *digital* pada perangkat *mobile* yang berfungsi untuk menampilkan pesan atau pemberitahuan secara real-time kepada pengguna tanpa perlu membuka aplikasi terlebih dahulu. Notifikasi dikirimkan melalui server aplikasi dan akan muncul pada layar perangkat pengguna dalam bentuk pesan singkat, pop-up, atau ikon pemberitahuan. Pada sistem operasi mobile modern, seperti Android, push notification telah menjadi mekanisme penting dalam meningkatkan interaksi pengguna terhadap aplikasi, karena pesan dapat diterima meskipun aplikasi tidak sedang dijalankan.

Secara konsep, push notification dirancang untuk memberikan informasi yang bersifat cepat, relevan, dan terarah. Berbeda dengan pesan konvensional seperti SMS atau email, notifikasi dikirim langsung oleh sistem aplikasi melalui layanan perantara (*notification service*), sehingga penyampaian pesan dapat dilakukan tanpa biaya tambahan bagi pengguna. Notifikasi juga dapat dikustomisasi sesuai kebutuhan, misalnya pesan teks, suara, getaran, atau tampilan gambar tertentu, sehingga memberikan pengalaman interaktif yang lebih personal kepada pengguna. Fungsi utama push notification adalah sebagai sarana pengingat, penyampai informasi penting, dan pemicu keterlibatan pengguna (*user engagement*). Dalam konteks layanan publik dan aplikasi berbasis kesehatan, push notification banyak dimanfaatkan sebagai pengingat aktivitas, informasi jadwal, atau peringatan tertentu yang berkaitan dengan pelayanan kesehatan masyarakat. Pengguna tidak hanya mendapatkan informasi secara cepat, tetapi juga dapat diarahkan untuk melakukan tindakan tertentu melalui notifikasi yang ditampilkan, seperti membuka aplikasi, melakukan konfirmasi, atau mengikuti kegiatan tertentu.

Dari perspektif interaksi pengguna, push notification juga memiliki manfaat dalam meningkatkan retensi pengguna (*user retention*), yaitu kemampuan aplikasi untuk membuat pengguna tetap aktif menggunakan layanan. Aplikasi yang memberikan notifikasi secara tepat waktu, informatif, dan sesuai kebutuhan pengguna cenderung memiliki tingkat penggunaan yang lebih tinggi dibandingkan aplikasi yang tidak memberikan notifikasi. Hal tersebut disebabkan karena notifikasi berperan sebagai pengingat dan penghubung antara pengguna dan sistem aplikasi dalam kurun waktu tertentu. Dalam pengembangan aplikasi mobile, penggunaan fitur push notification harus memperhatikan relevansi dan frekuensi pesan yang dikirimkan kepada pengguna. Notifikasi yang terlalu sering, tidak relevan, atau dianggap mengganggu justru dapat menurunkan minat pengguna terhadap aplikasi. Oleh karena itu, penerapan notifikasi perlu mempertimbangkan kebutuhan informasi pengguna, waktu pengiriman pesan, serta tujuan sistem yang dibangun. Ketepatan informasi menjadi prinsip penting agar notifikasi memberikan manfaat nyata dan tidak mengurangi kenyamanan pengguna.

Dalam sistem informasi donor darah, fitur *push notification* tidak hanya berfungsi sebagai pengingat jadwal donor, tetapi juga sebagai mekanisme peringatan dini terhadap kebutuhan darah. Pada sistem Donor Darah Online, rumah sakit dapat mengajukan permintaan darah melalui aplikasi ketika stok darah tidak mencukupi, kemudian sistem akan mengirimkan notifikasi kepada pendonor terdaftar berdasarkan kedekatan lokasi sehingga proses pencarian donor dapat dilakukan secara cepat dan terarah [11]. Penelitian di PMI Kota

Palembang juga menunjukkan bahwa sistem manajemen donor darah berbasis web yang dilengkapi notifikasi mampu menginformasikan kondisi stok darah rendah kepada pendonor dan masyarakat, sehingga mendorong partisipasi donor serta meningkatkan kesiapsiagaan PMI dalam menghadapi situasi darurat [9]. Studi internasional terkait aplikasi bank darah digital, seperti *Life Stream – A Blood Bank Application*, menegaskan bahwa integrasi *push notification* secara instan dan terpersonalisasi mampu meningkatkan keterlibatan pengguna melalui pembaruan informasi kegiatan donor, kampanye, dan kebutuhan darah secara real-time, meskipun pengaturan frekuensi notifikasi tetap perlu diperhatikan untuk menghindari *notification overload* [16]. Secara umum, penerapan *push notification* menjadikan komunikasi antara sistem dan pengguna lebih proaktif, sehingga informasi penting terkait donor darah dapat disampaikan secara cepat, tepat, dan personal melalui perangkat mobile.

2.5 Metode Pengembangan Sistem

Metode pengembangan sistem merupakan pendekatan terstruktur yang digunakan untuk merancang, membangun, menguji, dan mengimplementasikan perangkat lunak agar sesuai dengan kebutuhan pengguna serta mudah dipelihara. Pemilihan metode pengembangan sangat memengaruhi kualitas sistem, durasi pengembangan, kemampuan tim dalam merespons perubahan, serta efektivitas kolaborasi. Secara umum, metode pengembangan perangkat lunak terbagi menjadi dua kategori, yaitu metode tradisional (*sequential*) dan metode adaptif (*iterative*) [17].

Metode tradisional seperti *Waterfall* menggunakan pendekatan linier, di mana setiap tahapan pengembangan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Pendekatan ini sesuai untuk sistem dengan kebutuhan yang relatif stabil, namun kurang fleksibel ketika terjadi perubahan kebutuhan di tengah proses pengembangan [18]. Sebaliknya, metode adaptif (*iterative*) menekankan pengembangan yang dilakukan secara bertahap melalui siklus iterasi, di mana setiap iterasi mencakup pengembangan, pengujian, dan evaluasi berdasarkan umpan balik pengguna sehingga memungkinkan penyesuaian kebutuhan secara lebih cepat dan berkelanjutan [17].

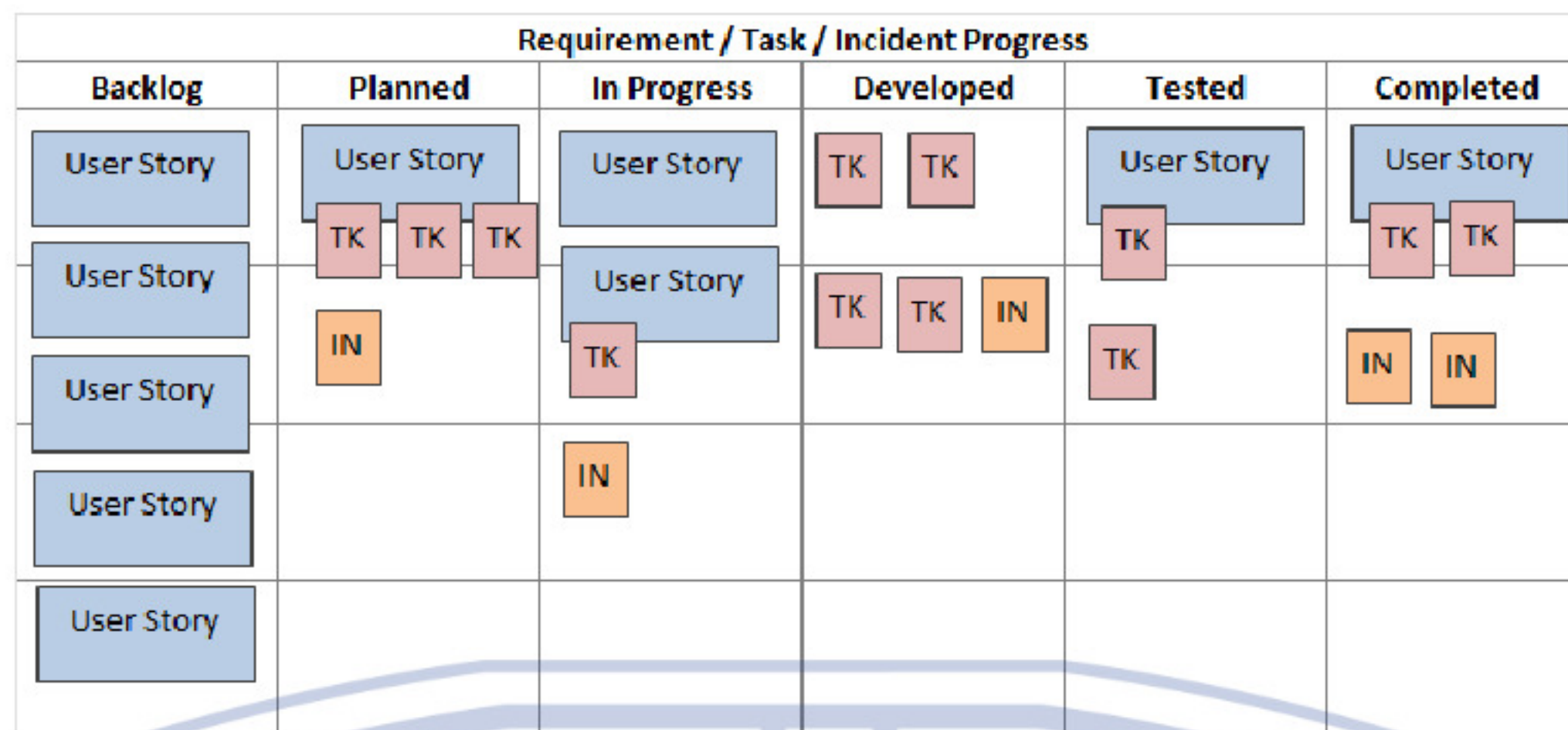
Salah satu pendekatan adaptif yang digunakan dalam pengembangan perangkat lunak adalah Agile. Agile menekankan pengembangan secara iteratif dan inkremental, kemampuan merespons perubahan, kolaborasi dengan pemangku kepentingan, serta evaluasi berdasarkan umpan balik pengguna. Agile tidak selalu harus diterapkan menggunakan satu kerangka kerja tertentu karena tingkat agility suatu proses pengembangan juga dipengaruhi oleh praktik yang dipilih dan konteks proyek yang dijalankan [19], [20], [21]. Dalam pengembangan aplikasi mobile, pendekatan Agile dinilai sesuai karena kebutuhan,

antarmuka, dan fungsi aplikasi dapat mengalami perubahan selama proses pengembangan dan pengujian. Pengembangan secara bertahap memungkinkan fitur aplikasi diperiksa, diperbaiki, dan disesuaikan berdasarkan hasil evaluasi pengguna [20], [22]. Penelitian empiris juga menunjukkan bahwa penerapan Agile memiliki hubungan positif dengan efisiensi proyek dan kepuasan pemangku kepentingan [23].

Pada penelitian ini, Agile digunakan sebagai pendekatan pengembangan sistem secara umum dan tidak diterapkan menggunakan kerangka kerja Kanban. Proses pengembangan dikelompokkan ke dalam kegiatan planning, design, development, testing, deployment, dan review. Kanban Board digunakan sebagai alat bantu pada tahap perencanaan untuk mencatat dan memvisualisasikan daftar pekerjaan yang diperlukan selama pengembangan aplikasi.

2.6 Metode Agile

Agile merupakan pendekatan pengembangan perangkat lunak yang menekankan proses iteratif dan inkremental. Sistem dikembangkan secara bertahap dalam bagian-bagian pekerjaan yang lebih kecil agar hasil pengembangan dapat diperiksa, diuji, serta disesuaikan berdasarkan perubahan kebutuhan dan umpan balik pengguna. Pendekatan ini tidak menetapkan satu rangkaian praktik yang harus diterapkan secara seragam pada setiap proyek karena pemilihan praktik Agile dapat disesuaikan dengan karakteristik, kebutuhan, dan kondisi proyek yang dijalankan [19], [20], [21]. Melalui pengembangan secara bertahap, tim dapat melakukan perbaikan terhadap fitur yang belum sesuai tanpa harus menunggu seluruh sistem selesai dikembangkan. Kolaborasi dan umpan balik dari pengguna menjadi bagian penting untuk memastikan bahwa sistem yang dibangun tetap relevan dengan kebutuhan pengguna. Dengan demikian, Agile memberikan fleksibilitas bagi tim untuk merespons perubahan selama proses pengembangan berlangsung [20], [23].



Gambar 2.1 Visualisasi Metode Agile Menggunakan Kanban[23]

Salah satu alat bantu yang dapat digunakan untuk mendukung pengelolaan pekerjaan dalam pengembangan perangkat lunak adalah Kanban Board. Kanban Board memvisualisasikan pekerjaan menggunakan kartu tugas dan kolom status yang menunjukkan posisi setiap pekerjaan dalam alur pengembangan. Penggunaan visualisasi tersebut dapat membantu tim memahami daftar pekerjaan, mengetahui status pekerjaan, dan mengendalikan aliran aktivitas pengembangan [19]. Penelitian mengenai penerapan Kanban dalam rekayasa perangkat lunak menunjukkan bahwa manfaat yang paling banyak didukung oleh bukti empiris mencakup peningkatan visibilitas pekerjaan, pengendalian aktivitas dan tugas, pengelolaan aliran pekerjaan, serta waktu penyelesaian yang lebih baik [23]. Pengelolaan aliran pekerjaan juga dapat membantu tim mengenali permasalahan, mengevaluasi proses, dan menentukan penyesuaian yang diperlukan selama pengembangan perangkat lunak [21]. Dalam penelitian ini, *Kanban Board* digunakan sebagai alat bantu untuk mencatat dan memvisualisasikan pekerjaan pada tahap perencanaan. *Kanban Board* yang digunakan terdiri atas bagian backlog serta kolom *To Do*, *Doing*, *Review*, dan *Done*.

Berdasarkan teori dan prinsip Agile, tim pengembang aplikasi donor darah berbasis Android di PMI Desa Hiliweto Gido memulai proses dengan identifikasi masalah dengan menggunakan metode Agile diterapkan untuk mengatasi berbagai masalah yang dihadapi oleh PMI, seperti ketidakstabilan informasi jadwal donor darah dan penggunaan kartu pendonor fisik yang rentan hilang atau rusak. Tahapan-tahapan dalam Agile yang dilakukan secara iteratif, meliputi:

1. **Planning (Perencanaan):** *Planning* merupakan kegiatan awal untuk mengidentifikasi permasalahan, tujuan pengembangan, kebutuhan pengguna, serta pekerjaan yang diperlukan dalam membangun aplikasi. Hasil identifikasi kebutuhan disusun menjadi daftar kebutuhan dan pekerjaan berdasarkan tingkat prioritasnya. Daftar tersebut

digunakan sebagai dasar dalam menentukan pekerjaan yang akan dilaksanakan pada tahap berikutnya.

2. *Design* (Perancangan): *Design* merupakan kegiatan merancang solusi sistem yang meliputi struktur sistem, arsitektur aplikasi, basis data, alur proses, serta antarmuka pengguna. Rancangan dapat diperbaiki atau disesuaikan apabila ditemukan kebutuhan baru atau ketidaksesuaian selama proses pengembangan dan evaluasi aplikasi.
3. *Development* (Implementasi): *Development* merupakan kegiatan mengimplementasikan kebutuhan dan rancangan sistem ke dalam bentuk aplikasi. Fitur aplikasi dikembangkan secara bertahap berdasarkan daftar pekerjaan dan prioritas yang telah ditentukan. Setelah suatu fitur selesai dikembangkan, fitur tersebut diperiksa dan diuji. Apabila ditemukan kesalahan atau ketidaksesuaian, fitur dikembalikan ke proses pengembangan untuk diperbaiki sebelum dinyatakan selesai [21], [23].
4. *Testing* (Pengujian): *Testing* dilakukan untuk memastikan bahwa setiap fungsi aplikasi berjalan sesuai dengan kebutuhan dan hasil yang diharapkan. Pengujian dilakukan setelah fitur selesai diimplementasikan sehingga kesalahan fungsi, kesalahan validasi, atau ketidaksesuaian keluaran dapat ditemukan dan diperbaiki lebih awal. Dalam penelitian ini, pengujian fungsional dilakukan menggunakan *Black Box Testing*.
5. *Deployment* (Penerapan): *Deployment* merupakan kegiatan penerapan aplikasi setelah fungsi utama sistem selesai dikembangkan dan telah melalui proses pengujian. Pada tahap ini, aplikasi didistribusikan, diinstal, dan diperkenalkan kepada pengguna atau pihak mitra. Penerapan aplikasi juga digunakan untuk memperoleh tanggapan mengenai penggunaan aplikasi dalam lingkungan operasional sebenarnya.
6. *Review* (Evaluasi): *Review* merupakan kegiatan evaluasi terhadap hasil pengembangan dan penggunaan aplikasi. Evaluasi dilakukan dengan memeriksa kesesuaian fungsi aplikasi terhadap kebutuhan sistem serta mempertimbangkan umpan balik dari pengguna dan pihak PMI Desa Hiliweto Gido. Apabila ditemukan fungsi yang belum sesuai, hasil evaluasi digunakan sebagai dasar untuk melakukan perbaikan pada bagian aplikasi yang berkaitan. Setelah perbaikan selesai, fungsi tersebut kembali diperiksa untuk memastikan bahwa hasilnya telah sesuai dengan kebutuhan pengguna [23].

2.7 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem berdasarkan kebutuhan dan spesifikasi tanpa memperhatikan struktur internal maupun kode program. Dalam metode ini, sistem diuji dari

sudut pandang pengguna dengan cara memberikan berbagai input dan kemudian mengamati output yang dihasilkan. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa setiap fungsi dalam sistem berjalan sesuai dengan yang diharapkan. Pengujian *black box* mampu mengidentifikasi berbagai kesalahan, seperti fungsi yang tidak berjalan dengan benar, kesalahan pada antarmuka, kesalahan validasi input, serta ketidaksesuaian antara output yang dihasilkan dengan spesifikasi sistem. Selain itu, metode ini memiliki kelebihan karena penguji tidak perlu memahami bahasa pemrograman yang digunakan, serta mampu merepresentasikan pengalaman pengguna secara langsung dalam menggunakan sistem [24].

Pada penelitian ini, metode *black box testing* diterapkan menggunakan teknik *equivalence partitioning*. Teknik ini digunakan untuk membagi data input ke dalam beberapa kelompok atau kelas ekuivalensi, yaitu data valid dan data tidak valid. Setiap data dalam satu kelompok diasumsikan akan diproses dengan cara yang sama oleh sistem, sehingga pengujian cukup dilakukan dengan menggunakan satu data perwakilan dari setiap kelompok. Dengan demikian, teknik ini dapat mengurangi jumlah *test case* tanpa mengurangi efektivitas pengujian [24]. Adapun tahapan pengujian *black box testing* dengan teknik *equivalence partitioning* dalam penelitian ini, yaitu sebagai berikut:

1. Mengumpulkan data halaman yang memiliki domain masukan
Tahap pertama dilakukan dengan mengidentifikasi dan mengumpulkan halaman pada sistem yang memiliki domain masukan. Domain masukan merupakan seluruh input yang terdapat pada sistem yang akan diuji. Tahap ini bertujuan untuk menentukan bagian sistem yang menjadi objek pengujian.
2. Mengumpulkan data domain masukan
Setelah halaman ditentukan, tahap selanjutnya adalah mengumpulkan data mengenai domain masukan dari setiap halaman. Data yang dikumpulkan berupa kondisi input, baik valid maupun tidak valid, yang akan digunakan sebagai dasar dalam proses pengujian.
3. Membuat partisi (*equivalence partitioning*)
Tahap berikutnya adalah membagi domain masukan ke dalam kelas-kelas ekuivalensi. Setiap kelas merepresentasikan sekumpulan data yang memiliki perlakuan yang sama oleh sistem. Partisi ini terdiri dari kelas data valid dan tidak valid, yang akan digunakan sebagai dasar dalam pembuatan skenario pengujian.
4. Membuat *test case*

Setelah partisi terbentuk, tahap selanjutnya adalah menyusun *test case*. Test case dibuat dengan mengombinasikan data dari masing-masing kelas ekuivalensi. Setiap test case berisi langkah pengujian, data input, serta hasil yang diharapkan (*expected result*).

5. Melakukan pengujian (*testing*)

Tahap terakhir adalah menjalankan seluruh test case yang telah dibuat pada sistem. Hasil pengujian kemudian dibandingkan antara *expected result* dan *actual result*. Jika hasil sesuai maka dinyatakan *pass*, dan jika tidak sesuai maka dinyatakan *fail*.

6. Evaluasi hasil pengujian dan perbaikan

Setelah pengujian dilakukan, hasil dianalisis untuk mengetahui adanya kesalahan atau ketidaksesuaian pada sistem. Apabila ditemukan *test case* yang gagal, maka dilakukan perbaikan terhadap sistem, kemudian dilakukan pengujian ulang hingga sistem berjalan sesuai dengan spesifikasi yang diharapkan.

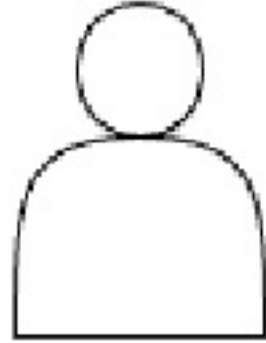
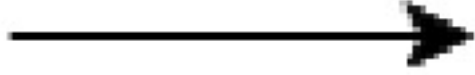
2.8 Use-Case Diagram

Use Case Diagram merupakan salah satu diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan fungsi-fungsi yang disediakan oleh sistem serta interaksi antara pengguna dan sistem. Use case diagram digunakan untuk memodelkan kebutuhan fungsional sistem dengan menampilkan aktor yang terlibat dan layanan yang dapat diberikan oleh sistem kepada aktor tersebut. Use case diagram menjelaskan bagaimana pengguna berinteraksi dengan sistem melalui serangkaian *use case* yang merepresentasikan proses atau aktivitas utama sistem. Dengan menggunakan use case diagram, pengembang dapat memahami ruang lingkup sistem, fungsi-fungsi yang dapat dijalankan, serta hubungan antara aktor dan sistem secara lebih terstruktur [25].

Dalam perancangan sistem, use case diagram berperan sebagai alat bantu untuk menggambarkan struktur fungsional sistem secara umum sebelum dilakukan perancangan lebih detail. Diagram ini membantu mengidentifikasi proses-proses utama yang dapat dilakukan oleh sistem dan pihak-pihak yang terlibat di dalamnya, sehingga dapat meminimalkan kesalahan dalam tahap pengembangan selanjutnya. Pada use case diagram digunakan sejumlah simbol untuk merepresentasikan elemen-elemen sistem. Setiap simbol memiliki makna tertentu yang digunakan untuk memperjelas interaksi antara aktor dan use case agar diagram dapat dipahami dengan jelas dan akurat. Simbol-simbol yang digunakan pada use case diagram ditunjukkan pada Tabel 2.1

Tabel 2.1 Use Case Diagram [25]

No.	Simbol	Nama	Keterangan
-----	--------	------	------------

1.		<i>Actor</i>	Entitas (orang, sistem lain, atau organisasi) yang berinteraksi langsung dengan sistem untuk menjalankan suatu <i>use case</i> .
2.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen lain yang bergantung padanya.
3.		<i>Generalization</i>	Hubungan pewarisan dimana objek anak (<i>descendant</i>) mewarisi perilaku dan struktur data dari objek induk (<i>ancestor</i>).
4.		<i>Include</i>	Menspesifikasikan bahwa suatu <i>use case</i> selalu menyertakan (<i>include</i>) perilaku dari <i>use case</i> lain secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa suatu <i>use case</i> dapat memperluas perilaku dari <i>use case</i> lain pada kondisi atau titik tertentu.
6.		<i>Association</i>	Hubungan yang menunjukkan adanya interaksi atau komunikasi antara aktor dengan <i>use case</i> dalam sistem.
7.		<i>Use Case</i>	Deskripsi dari serangkaian aksi yang dilakukan sistem untuk menghasilkan suatu hasil yang bernilai dan terukur bagi aktor.

2.9 Model Standar ISO/IEC 25010

ISO/IEC 25010, yang merupakan bagian dari seri standar SQuaRE (*Systems and software Quality Requirements and Evaluation*), adalah kerangka kerja internasional yang mendefinisikan model kualitas untuk produk perangkat lunak dan sistem komputer. Model ini merupakan pembaruan dari standar pendahulunya (ISO/IEC 9126) dengan menambahkan

cakupan karakteristik yang lebih relevan terhadap perkembangan teknologi modern, seperti aplikasi *mobile* [26]. Dalam praktik rekayasa perangkat lunak, evaluasi kualitas sistem tidak hanya diukur dari fitur yang tersedia (fungsional), tetapi juga dari bagaimana sistem tersebut beroperasi (non-fungsional). Seringkali kebutuhan non-fungsional diabaikan dalam proses perancangan, meskipun sangat krusial untuk menunjang kenyamanan dan pengoptimalan operasional sistem [27], [28]. ISO/IEC 25010 menyediakan landasan yang komprehensif untuk mendefinisikan dan mengevaluasi spesifikasi kebutuhan non-fungsional tersebut.

Kerangka ini membagi kualitas operasional produk perangkat lunak ke dalam tujuh karakteristik non-fungsional utama, yaitu:

1. *Usability* (Kebergunaan)

Mendeskripsikan sejauh mana antarmuka sistem dapat digunakan oleh pengguna secara mudah, efisien, dan memuaskan [29], [30]. Penilaian kebutuhan non-fungsional pada dimensi ini mencakup kemudahan aplikasi untuk dipelajari oleh pengguna awam (*learnability*), kemudahan dioperasikan (*operability*), perlindungan serta mitigasi dari kesalahan input pengguna (*user error protection*), dan estetika desain antarmuka (*user interface aesthetics*).

2. *Performance Efficiency* (Efisiensi Kinerja)

Mengukur seberapa efisien perangkat lunak merespons instruksi menggunakan sumber daya yang ada di bawah kondisi operasional tertentu. Indikator evaluasi non-fungsional pada dimensi ini mencakup waktu respons aplikasi terhadap suatu perintah pengguna (*time behaviour*), seberapa besar kapasitas sistem dalam menangani data, serta pemanfaatan sumber daya sistem seperti memori dan baterai perangkat *mobile* (*resource utilization*) [29], [30].

3. *Reliability* (Keandalan)

Mengukur sejauh mana aplikasi dapat mempertahankan tingkat kinerjanya tanpa mengalami gangguan atau *crash* dalam rentang waktu penggunaan yang spesifik. Indikator keandalan meliputi ketersediaan sistem untuk diakses (*availability*), kelancaran aplikasi saat terjadi interupsi seperti hilangnya koneksi internet (*fault tolerance*), dan kemampuan pemulihan sistem jika terjadi kegagalan (*recoverability*) [30].

4. *Security* (Keamanan)

Menilai kemampuan perangkat lunak dalam melindungi informasi dan data pribadi pengguna sehingga pihak yang tidak memiliki hak akses sistem tidak dapat membaca atau memodifikasinya. Sub-karakteristik keamanan mencakup perlindungan

kerahasiaan data (*confidentiality*), keutuhan data riwayat pengguna (*integrity*), dan akuntabilitas sistem [30].

5. *Compatibility* (Kompatibilitas)

Menilai kemampuan perangkat lunak untuk saling bertukar informasi secara efektif atau menjalankan fungsinya tanpa mengganggu lingkungan perangkat keras maupun perangkat lunak lainnya. Pada aplikasi *mobile*, dimensi ini sangat krusial untuk memastikan sistem dapat berjalan stabil di berbagai variasi dan versi sistem operasi Android yang digunakan oleh pendonor [30].

6. *Maintainability* (Pemeliharaan)

Dimensi ini mengukur kemudahan produk perangkat lunak untuk dimodifikasi, diperbarui, atau diadaptasi terhadap kebutuhan di masa depan. Karakteristik ini meliputi arsitektur kode yang terstruktur (*modularity*), kemudahan modifikasi (*modifiability*), dan kemudahan pengujian fitur baru (*testability*) oleh pengembang [29], [30].

7. *Portability* (Portabilitas)

Portabilitas merujuk pada kemudahan sistem atau komponen perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain. Dalam konteks aplikasi *mobile*, ini mencakup kemudahan aplikasi untuk diunduh, diinstal (*installability*), serta dicopot oleh pengguna tanpa meninggalkan galat (*error*) pada perangkat keras [30].

Pendekatan menggunakan tujuh karakteristik non-fungsional dari ISO/IEC 25010 ini memastikan bahwa perancangan aplikasi donor darah tidak hanya berfokus pada kelengkapan fitur, melainkan juga menutupi area operasional yang perlu ditingkatkan seperti stabilitas, efisiensi, dan keamanan [28]. Metode ini memberikan metrik evaluasi berstandar internasional untuk menjamin bahwa aplikasi memiliki performa yang efisien, aman, dan mudah digunakan untuk menunjang kelancaran operasional administrasi secara optimal [30].

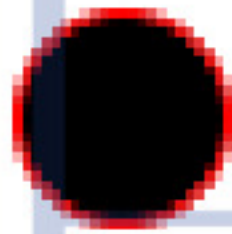
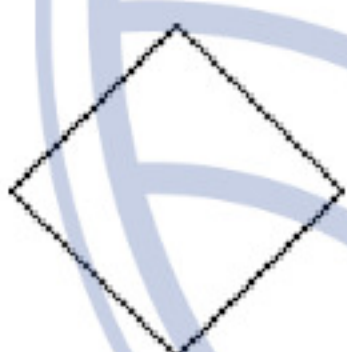
2.10 Activity Diagram

Activity Diagram merupakan salah satu diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan alur aktivitas atau proses bisnis yang terjadi di dalam sistem secara berurutan. Activity diagram memvisualisasikan langkah-langkah aktivitas dari awal hingga akhir, termasuk urutan proses, keputusan, serta kondisi yang mungkin terjadi selama sistem berjalan. Diagram ini digunakan untuk memberikan gambaran yang lebih rinci mengenai perilaku sistem dalam menjalankan suatu fungsi. Activity diagram berfungsi sebagai pelengkap use case diagram dengan menjelaskan detail alur proses internal dari setiap use case yang telah didefinisikan sebelumnya. Diagram ini menunjukkan bagaimana sistem merespons aksi yang dilakukan oleh aktor melalui

serangkaian aktivitas yang terstruktur, sehingga memudahkan pengembang dalam memahami logika proses dan aliran kerja sistem [31].

Dalam perancangan sistem, activity diagram digunakan untuk menggambarkan urutan aktivitas, titik keputusan, serta transisi antar aktivitas yang terjadi dalam suatu proses. Dengan adanya activity diagram, implementasi sistem dapat disesuaikan dengan alur kerja yang telah dirancang, sehingga sistem yang dibangun dapat berjalan sesuai dengan kebutuhan fungsional yang diharapkan [31]. Berikut merupakan simbol-simbol pada activity diagram:

Tabel 2.2 Simbol-Simbol Activity Diagram [31]

No.	Simbol	Nama	Keterangan
1.		<i>Initial</i>	Menunjukkan titik awal atau keadaan awal dari suatu <i>activity diagram</i> sebelum proses atau aktivitas dijalankan.
2.		<i>Dependency</i>	Menunjukkan titik akhir dari suatu <i>activity diagram</i> yang menandakan bahwa seluruh aktivitas telah selesai.
3.		<i>Activity</i>	Merepresentasikan suatu aktivitas atau aksi yang dilakukan dalam alur proses sistem.
4.		<i>Decision</i>	Digunakan untuk menunjukkan percabangan alur berdasarkan suatu kondisi atau keputusan tertentu (biasanya menghasilkan pilihan Ya/Tidak atau Benar/Salah).

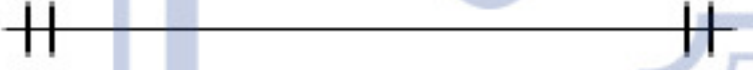
2.11 Entity Relational Diagram

Entity Relationship Diagram (ERD) merupakan model konseptual yang digunakan untuk menggambarkan struktur data dan hubungan antar data dalam suatu sistem. ERD memodelkan data dalam bentuk entitas, atribut, dan relasi yang merepresentasikan objek-objek nyata beserta keterkaitannya di dalam domain sistem yang dirancang. ERD berperan penting dalam tahap perancangan basis data karena mampu menjelaskan kebutuhan data secara terstruktur dan mudah dipahami [32]. ERD digunakan sebagai dasar dalam merancang struktur logis basis data sebelum diimplementasikan ke dalam sistem manajemen basis data relasional. Dengan menggunakan ERD, perancang sistem dapat mengidentifikasi entitas utama, menentukan atribut yang dimiliki setiap entitas, serta mendefinisikan

hubungan antar entitas secara jelas. Diagram ini membantu memastikan bahwa data yang disimpan sesuai dengan kebutuhan sistem dan mampu mendukung proses pengolahan data secara konsisten [32].

Dalam perancangan sistem, ERD berfungsi sebagai jembatan antara kebutuhan pengguna dan implementasi teknis basis data. ERD memberikan gambaran menyeluruh mengenai struktur data yang bersifat statis dan menjadi landasan yang kuat untuk proses transformasi ke dalam model basis data relasional. Dengan demikian, penggunaan ERD dapat meminimalkan kesalahan perancangan data serta meningkatkan kesesuaian antara sistem yang dibangun dan kebutuhan informasi yang diharapkan.

Tabel 2.3 Kardinalitas Hubungan ERD [32]

No.	Notasi	Simbol	Keterangan
1.		<i>One-one</i>	Hubungan dimana satu objek pada suatu entitas hanya berhubungan dengan satu objek pada entitas lainnya, dan sebaliknya.
2.		<i>one-to-many</i>	Hubungan dimana satu objek pada suatu entitas dapat berhubungan dengan banyak objek pada entitas lain, tetapi satu objek pada entitas kedua hanya berhubungan dengan satu objek pada entitas pertama.
3.		<i>many-to-one</i>	Hubungan dimana banyak objek pada suatu entitas berhubungan dengan satu objek pada entitas lain.
4.		<i>many-to-many</i>	Hubungan dimana banyak objek pada suatu entitas dapat berhubungan dengan banyak objek pada entitas lainnya.