

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Banyak tipe orang yang menggunakan sepatu, yang dipakai untuk fashion dan sebagai sarana melindungi kaki dari panas, benda tajam, dan kotoran. Tidak mungkin mencegah sepatu menjadi kotor saat dipakai sepatu yang sering dipakai juga bisa menjadi lembap karena keringat dan menimbulkan bau tidak sedap dari munculnya mikroorganisme. Sepatu perlu dibersihkan dan dikeringkan secara menyeluruh agar tetap terjaga kondisinya karena juga rentan terkena air hujan jika dipakai untuk berjalan kaki atau mengendarai sepeda motor. Secara tradisional, sepatu dikeringkan dengan cara dijemur di bawah terik sinar matahari dan diangin-anginkan, meskipun cuaca yang tidak menentu terkadang membuat proses ini menjadi lebih sulit[3].

Di musim hujan dan malam hari, saat matahari tidak bersinar, pengering sepatu bisa menjadi pengganti yang berguna. Pengering sepatu ini akan menjadi gadget yang bekerja selaras dengan kemajuan teknologi modern jika dipasangkan dengan modul *bluetooth*[4].

Laporan tugas akhir Sistem Kontrol Pengering Sepatu Berbasis Android yang ditulis oleh Muhammad Syarif Fauzi Fadilah (2022) melakukan penelitian ini dan menemukan bahwa sistem kendali perangkat elektronik merupakan bidang yang menarik untuk dikembangkan dan sedang digemari. Salah satu kemajuan yang bisa dilakukan untuk memudahkan pekerjaan

manusia adalah dengan memanfaatkan sebaik-baiknya teknologi yang sudah ada, khususnya sistem operasi android. Pengendalian peralatan elektronik saat ini semakin mudah karena sebagian besar perangkat elektronik yang digunakan dihubungkan dengan ponsel pintar (*smartphone*) yang merupakan salah satu barang yang paling sering digunakan masyarakat dalam aktivitas sehari-hari[5].

2.2 Landasan Teori

2.2.1 Arduino IDE



Gambar 2. 1 Arduino IDE

(Sumber: <https://images.app.goo.gl/Mu2Fjv5rnqqQw74D8>)

Arduino IDE (*Integrated Development Environment*) menurut McRoberts (2010) merupakan perangkat lunak bebas yang digunakan untuk menulis kode dalam bahasa pemrograman (C) yang mampu dipahami oleh arduino. IDE biasanya digunakan untuk membuat program komputer, yang pada dasarnya adalah sekumpulan instruksi berurutan yang diunggah ke arduino[6].

ATmega328 adalah dasar dari papan mikrokontroler Arduino Uno (*datasheet*). Memiliki total 14 pin input, termasuk osilator kristal 16 MHz, 6 pin *input* digital yang dapat digunakan untuk *output* PWM, 6

pin *input* analog, port USB, colokan listrik, header ICSP, dan tombol reset. Cukup sambungkan papan Arduino Uno ke komputer melalui kabel USB, atau gunakan adaptor AC-DC atau baterai untuk menyalakannya, untuk mengaktifkan mikrokontroler dan membuatnya dapat digunakan[7].

Pemrograman dengan Java adalah dasar dari Arduino IDE. Selain itu, Arduino IDE dilengkapi dengan pustaka C/C++ yang disebut *Wiring* yang menyederhanakan operasi *input* dan *output*. Arduino IDE ini dibuat dengan mendesain ulang perangkat lunak pemrosesan untuk membuat Arduino IDE yang disesuaikan untuk pemrograman Arduino. Sketsa merupakan program yang dibuat dengan *Software* Arduino (IDE). Editor teks digunakan untuk menulis sketsa, yang kemudian disimpan sebagai file dengan ekstensi, *ino*. Penulisan kode program menjadi lebih sederhana dengan editor teks perangkat lunak arduino, yang mencakup fungsi seperti menyalin, menempel, serta menghapus dan mengganti. Ada semacam jendela pesan hitam di perangkat lunak Arduino IDE yang menampilkan informasi status, termasuk peringatan kesalahan, mengkompilasi program, dan mengunggahnya. Di bagian paling bawah[8].

Papan mikrokontroler berbasis ATmega328 adalah Arduino UNO. Arduino UNO dilengkapi dengan osilator kristal 16 MHz, 6 input analog, 14 pin *input/output* digital (enam di antaranya dapat digunakan sebagai *output* PWM), port USB, colokan listrik, header ICSP, dan

tombol reset. Semua komponen yang diperlukan untuk mendukung mikrokontroler disertakan dalam Arduino UNO. Ini mungkin hanya dihubungkan ke komputer melalui kabel USB, adaptor AC ke DC, atau baterai untuk menyalakannya[9].

2.2.2 Sistem Operasi Android



Gambar 2. 2 Sistem Operasi Android

(Sumber: <https://images.app.goo.gl/6jtqqdvJ4ZRUBT5FA>)

Android merupakan sistem operasi berbasis Linux yang ditujukan untuk perangkat seluler dengan layar sentuh, seperti tablet dan ponsel pintar. Didirikan pada bulan Oktober 2003, Android, Inc. dipimpin oleh Chris White, kepala desain dan pengembangan antarmuka WebTV, Nick Sears, mantan wakil presiden T-Mobile, Rich Miner, pendiri Wildfire Communications, Inc, dan Andy Rubin, pendiri Danger, dengan tujuan menciptakan "perangkat seluler pintar yang lebih mengetahui lokasi dan preferensi penggunanya". Tujuan awal Android adalah untuk menciptakan sistem operasi canggih untuk kamera digital, namun setelah menjadi jelas bahwa tidak ada pasar yang cukup besar untuk perangkat semacam ini, pengembangan Android dialihkan ke

ruang ponsel pintar untuk bersaing dengan Symbian dan Windows Mobile[10].

Android adalah sistem operasi seluler berbasis Linux yang dirancang untuk tablet dan ponsel pintar. Pemrogram dapat membuat aplikasinya sendiri pada platform terbuka menggunakan Android di berbagai perangkat yang didukung Android[11].

2.2.3 Bluetooth



Gambar 2. 3 Bluetooth

(Sumber: <https://images.app.goo.gl/BbrnhKeazP1oLfeJ9>)

Teknologi nirkabel tambahan yang dirancang khusus untuk perangkat seluler adalah *bluetooth*. Berbeda dengan WiFi, *bluetooth* menggunakan standar yang berkaitan dengan spesifikasi. *bluetooth* beroperasi pada frekuensi 2,4 GHz dan mentransfer data dengan kecepatan 800 Kbps atau kurang, kurang dari 1 Mbps. Gadget *bluetooth* dapat berkomunikasi hingga 13 meter dengan peralatan lainnya. Standar baru yang dapat menempuh jarak sekitar 100 meter baru saja dirancang (tanpa penghalang)[12].

Teknologi baru yang dikenal sebagai *bluetooth* mulai mendapatkan popularitas. Perangkat listrik yang kita gunakan telah

mengalami perubahan besar karena teknologi ini. Melihat sekeliling, kita dapat melihat di mana *keyboard* dan komputer terhubung[13].

Nama *bluetooth* diambil dari nama seorang raja yang hidup pada pertengahan abad kesepuluh, Harald Blåtand, juga dikenal sebagai *buetooth* dalam bahasa Inggris. Denmark dan Norwegia bersatu dan diperintah oleh Harald Blåtand. Hal ini memunculkan istilah *bluetooth* untuk perangkat yang saling terhubung[14].

2.2.4 App Inventor



Gambar 2. 4 App Inventor

(Sumber: <https://images.app.goo.gl/cjX1diPB62RotmYN9>)

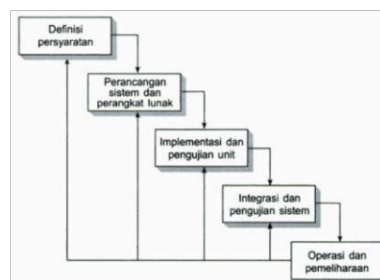
App Inventor adalah aplikasi online sumber terbuka yang kini dikelola oleh *Massachusetts Institute of Technology* (MIT). Ini pertama kali dibuat oleh Google. Dengan App Inventor, pemrogram komputer pemula dapat menulis perangkat lunak untuk sistem operasi *android*. Menggunakan antarmuka grafis yang mirip dengan *Scratch* dan StarLogo TNG, App Inventor memungkinkan pengguna membuat aplikasi *android* hanya dengan menyeret dan melepaskan komponen visual. Google menyelesaikan lingkungan pengembangan online

Google dan melakukan sembilan studi yang berkaitan dengan komputasi pendidikan untuk menciptakan App Inventor.

Anda tidak perlu mengetahui atau menggunakan banyak bahasa pemrograman untuk membuat aplikasi dasar dengan bantuan MIT App Inventor. Pengguna dapat menggunakan beberapa tata letak dan komponen yang tersedia untuk mengembangkan aplikasi android sesuai keinginan mereka.[15].

Karena memungkinkan pengguna untuk memanfaatkan, memeriksa, mengatur, dan menarik dan melepas blok simbol perintah untuk membuat dan mengembangkan aplikasi seluler yang dapat beroperasi pada sistem operasi Android, MIT App Inventor adalah alat pemrograman blok visual[16].

2.2.5 Metode Waterfall



Gambar 2. 5 Metode Waterfall

(Sumber: <https://images.app.goo.gl/5dk3KC2d8DsZ6RE76>)

Metode ini terdiri dari langkah-langkah yang diselesaikan secara berurutan dan linier, termasuk analisis persyaratan, desain, implementasi, pengujian, pengiriman, dan pemeliharaan. Pendekatan ini kurang fleksibel dalam menangani perubahan kebutuhan yang

mungkin timbul selama proses pembangunan, meskipun pendekatan ini sesuai untuk proyek dengan persyaratan yang stabil dan terdefinisi dengan baik.

2.2.6 Sparx Systems Enterprise Architect



Gambar 2. 6 Sparx Systems Enterprise Architect

(Sumber: <https://images.app.goo.gl/fKy5Ax9Yr8q9o8Pg9>)

Perusahaan perangkat lunak Australia Sparx Systems mengembangkan platform visual *Sparx Systems Enterprise Architect* untuk digunakan dalam desain dan pengembangan sistem perangkat lunak, pemodelan proses bisnis, simulasi, dan aplikasi pemodelan lainnya. Manfaat menggunakan *Sparx Systems Enterprise Architect* adalah sebagai berikut: Pertama, dengan menggunakan standar BPMN dan UML, *Sparx Systems Enterprise Architect* dapat memodelkan dan mensimulasikan proses bisnis dan aplikasi. Kedua Arsitek Perusahaan Sistem Sparx memiliki kemampuan untuk sepenuhnya memodelkan proses bisnis, menghubungkan data, aplikasi, dan bahaya, serta mempublikasikan hasil dalam berbagai bentuk. Di sektor industri, ketiga Arsitek Perusahaan Sistem Sparx adalah tolok ukur yang

diterima untuk analisis dan desain aplikasi menyeluruh (Sistem, Basis Data, dan Desain Antarmuka Pengguna). Keempat Arsitek Perusahaan Sistem *Sparx* sangat mudah beradaptasi dan disetujui sebagai instrumen yang membantu

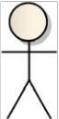
2.2.7 UML


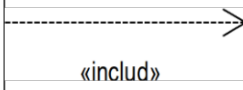
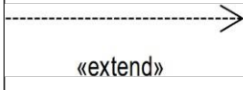
Istilah "*Unified Modeling Language*" (UML) mengacu pada teknik pemodelan visual yang digunakan dalam desain sistem berorientasi objek. Ini juga merupakan bahasa yang digunakan secara luas dalam visualisasi, desain, dan dokumentasi sistem perangkat lunak. Saat ini, UML banyak digunakan sebagai bahasa standar untuk penulisan cetak biru perangkat lunak.

A. Diagram Use Case

Berfungsi sebagai deskripsi fungsionalitas sistem yang diantisipasi dan menunjukkan bagaimana pengguna berinteraksi dengannya. Aktor adalah deskripsi entitas dari seseorang atau sistem yang melakukan tugas untuk sistem dalam suatu use case.

Tabel 2. 1 Use Case



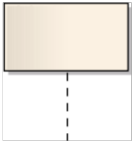
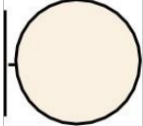
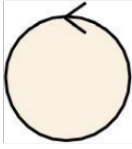
No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Mewakili peran pengguna, sistem atau alat ketika berkomunikasi dengan <i>use case</i> .

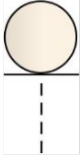



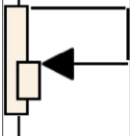
No	Simbol	Nama	Keterangan
2		<i>Use Case</i>	Abstraksi atau Interaksi antara Sistem dengan actor.
3		<i>Association</i>	Abstraksi dari penghubung antara actor dengan use case.
4		<i>Generalize</i>	Menunjukkan spesialisasi actor. Untuk dapat berpartisipasi dengan use case.
5		<i>Include</i>	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsional dari use case lain.
6		<i>Extend</i>	Menunjukkan bahwa suatu usecase merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi.

B. Diagram Sequence

Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu.

Tabel 2. 2 Diagram Sequence




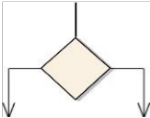
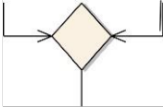

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Merepresentasikan entitas yang berada di luar sistem dan berinteraksi dengan sistem.
2		<i>Lifeline</i>	Menghubungkan objek selama <i>Sequence</i> (<i>message</i> dikirim atau diterima dan aktivitasnya).
3		<i>General</i>	Merepresentasikan entitas tunggal dalam <i>Sequence diagram</i> .
4		<i>Boundary</i>	Berupa tepi dari sistem, seperti <i>user interface</i> atau suatu alat yang berinteraksi dengan sistem yang lain.
5		<i>Control</i>	<i>Element</i> yang mengatur aliran dari informasi untuk sebuah skenario, Objek ini umumnya mengatur perilaku dan perilaku bisnis.

No	Simbol	Nama	Keterangan
6		<i>Entity</i>	Element yang bertanggung jawab menyimpan data atau informasi.
7		<i>Activation</i>	Suatu titik dimana sebuah objek mulai berpartisipasi didalam sebuah Sequence yang menunjukkan kapan sebuah objek mengirim atau menerima.
8		<i>Message</i>	Berfungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan.
9		<i>Message Entry</i>	Berfungsi untuk menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
10		<i>Self Message</i>	Sebuah pesan yang mendefinisikan komunikasi tertentu antara Lifeline.

C. Diagram Activity

Merupakan gambaran alir dari aktivitas- aktivitas didalam sisten yang berjalan.

Tabel 2. 3 Diagram Activity

No	Simbol	Nama	Keterangan
1		<i>Initial State</i>	Awal dimulainya suatu aliran kerja pada sebuah <i>Activity diagram</i> .
2		<i>Final State</i>	Bagian akhir dari suatu aliran kerja pada sebuah <i>Activity diagram</i> .
3		<i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
4		<i>Decision</i>	Berfungsi untuk menggambarkan pilihan kondisi dimana ada kembangkinan perbedaan transisi, untuk memastikan bahwa aliran kerja dapat mengalir ke lebih dari satu jalur.
5		<i>Merge</i>	Berfungsi untuk menghubungkan kembali aliran kerja yang sebelumnya telah dipecah oleh <i>Decision</i> .
6		<i>Partition</i>	Dapat digunakan untuk mengilustrasikan aktivitas yang dilakukan oleh <i>actor</i> yang berbeda.