

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Penelitian yang dilakukan oleh Syafrie Syahriel, dkk. (2021) dengan penelitian yang berjudul Perancangan Atm Raskin Berbasis Rfid dan *Internet Of Things* (IOT) Untuk Masyarakat Tidak Mampu. Dalam penelitian ini membuat ATM Beras dengan menggunakan RFID (*Radio Frequency Identification*) dan IOT (*Internet Of Things*), yang akan ditempatkan di kantor kelurahan. Dengan adanya perangkat ini, mempermudah pembuatan laporan yang tersimpan dalam sistem yang telah disiapkan. Setiap transaksi mesin ATM akan mengirimkan data kepada penanggung jawab[5].

Penelitian yang hampir sama juga dibuat oleh Safira Angelia Putri, dkk. (2022) dengan penelitian yang berjudul Rancang Bangun ATM Beras Raskin Dengan RFID (*Radio Frequency Identification*) E-KTP (Elektronik Kartu Tanda Penduduk) Menggunakan Teknik *Counter* Berbasis Arduino Uno. Dalam penelitian ini membuat sebuah desain mesin ATM untuk beras masyarakat menggunakan modul RFID (*Radio Frequency Identification*) sebagai identifikasi, Arduino Uno menjadi pusat kontrol utama, dan menerapkan teknik *counter* untuk membatasi penggunaan kartu. Hasilnya, mesin ini dapat beroperasi dengan efisien. Setelah proses pengambilan beras selesai, sistem akan mengirimkan notifikasi hasilnya yang ditampilkan di layar LCD (*Liquid Crystal Display*). Mesin ATM ini dirancang untuk memberi bantuan masyarakat membutuhkan untuk mengambil beras[6].

Pada penelitian lain yang hampir sama juga dibuat oleh Yuli Darni, dkk. (2023) dengan penelitian yang berjudul Pembangunan Ekonomi Islam Oleh Ybm Brilian: Program Zakat Dan Wakaf Atm Beras Di Masjid Raya Stabat. Pada penelitian ini melakukan aktivitas untuk memajukan perkembangan ekonomi Islam dengan menerapkan wakaf dan zakat sebagai sarana. Tidak hanya bersifat sebagai ibadah ritual, tetapi juga dalam aspek keadilan, sosial, dan kesejahteraan manusia. Salah satu contohnya adalah yang dilakukan oleh YBM Brilian dengan program Wakaf ATM Beras di Masjid Raya Stabat[7].

Selain penelitian yang diatas, penelitian yang hampir sama juga dibuat oleh Widya Permata, dkk. (2021) dengan penelitian yang berjudul Strategi Pengelolaan Atm Beras Dalam Perspektif Ekonomi Syariah (Studi Kasus Di Masjid Yamp Al-Muhajirin, Banjarmasin). Penelitian ini mengkaji langkah pengelolaan yang diterapkan pengurus Masjid YAMP Al-Muhajirin Banjarmasin untuk mengelola ATM beras serta strategi pengelolaannya dari perspektif ekonomi syariah. Tujuannya adalah untuk memahami langkah pengelolaan yang diterapkan pengurus Masjid YAMP Al-Muhajirin Banjarmasin dalam mengoperasikan ATM beras dan menganalisis pengelolaan tersebut dari sudut pandang ekonomi syariah. Metode yang digunakan yaitu kualitatif serta pendekatan studi kasus, dan lokasi penelitian dipilih di Masjid YAMP Al-Muhajirin Banjarmasin[8].

2.2 Landasan Teori

2.2.1 ATM Beras

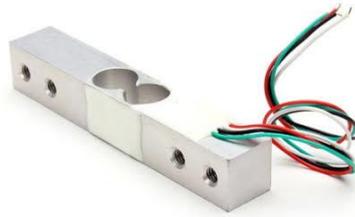
ATM beras adalah perangkat elektronik yang dirancang untuk menyimpan dan mendistribusikan beras secara otomatis dengan penggunaan mikrokontroler, RFID (*Radio Frequency Identification*) sebagai masukan data pengguna, motor servo sebagai pembuka dan penutup katup pada tabung penampung penyimpanan beras yang akan dikeluarkan sesuai dengan data yang ada pada kartu RFID (*Radio Frequency Identification*)[9].



Gambar 2. 1 ATM Beras

2.2.2 Sensor Load Cell

Load Cell adalah alat yang berguna untuk menghitung beban dengan cara merubah gaya atau tekanan mekanis menjadi sinyal listrik. Ketika ada barang diatas *load cell*, perangkat ini menghasilkan sinyal listrik yang senilai dengan kekuatan tekanan yang diterapkan [10].



Gambar 2. 2 Sensor *Load Cell*

Keterangan gambar :

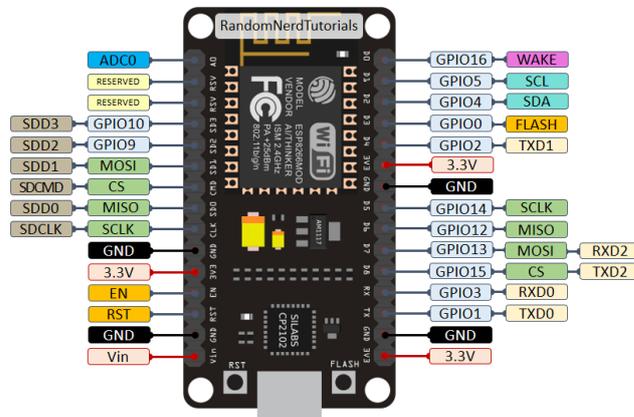
1. Kabel merah berfungsi sebagai *input* tegangan untuk sensor.
2. Kabel hitam berfungsi sebagai *input ground* untuk sensor.
3. Kabel hijau berfungsi sebagai *output* positif dari sensor.
4. Kabel putih berfungsi sebagai *output ground* dari sensor.

Prinsip kerja alat ini didasarkan pada perubahan tekanan dan resistansi. *Load Cell* terdiri dari elemen yang menghasilkan sinyal tegangan. Saat beban melewati sensor, akan menghasilkan sinyal elektrik karena tarikan dan tekanan yang dikenakan pada *strain gauge* yang ada di dalam load cell[10].

2.2.3 NodeMCU ESP8266

NodeMCU adalah sebuah perangkat keras yang dirancang dengan mendukung produk *Internet of Things*. NodeMCU merupakan perkembangan dari perangkat keras Arduino dengan tambahan fitur-fitur modern. Salah satu fitur utama NodeMCU adalah modul Wi-Fi yang terhubung pada sirkuit. Hal ini memungkinkan NodeMCU untuk

tersambung ke koneksi Wi-Fi tanpa perlu perangkat tambahan modul eksternal[11].



Gambar 2. 3 NodeMCU

2.2.4 Arduino IDE

Arduino IDE yaitu sebuah aplikasi untuk melakukan pemrograman pada Arduino. Melalui aplikasi ini, Arduino diprogram untuk menjalankan berbagai fungsi yang telah disusun menggunakan kode program. Arduino menggunakan bahasa program C yang telah dimodifikasi untuk mengatur operasinya[12].

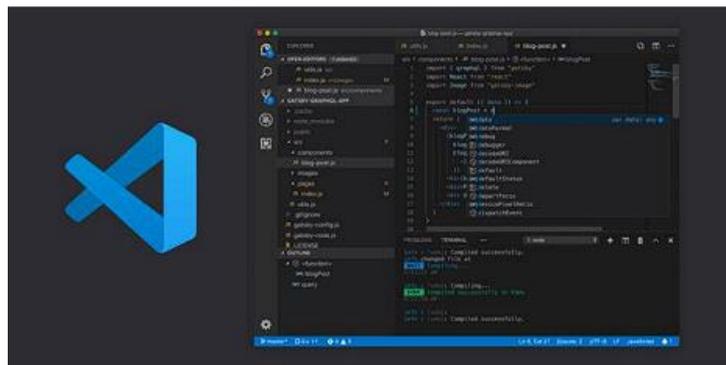


Gambar 2. 4 Arduino IDE

2.2.5 Visual Studio Code

Visual Studio Code (VS Code) yaitu editor yang dapat diandalkan yang dikembangkan *Microsoft*. Editor ini dirancang untuk berbagai OS (*Operating System*), termasuk *Linux*, *Mac*, dan *Windows*. *VS Code* mendukung langsung bahasa program seperti *Node.js*, *TypeScript*, *JavaScript*, dan berbagai bahasa program yang lain melalui *plugin* yang diunduh dari *marketplace Visual Studio Code*. *Plugin* tersebut mencakup bahasa seperti *Java*, *Python*, *Go*, *C#*, *C++*, dan lainnya[13].

Visual Studio Code (VS Code) merupakan perangkat lunak sumber terbuka, yang artinya kodenya dapat diubah serta bisa berkontribusi untuk pengembangan. Kode sumber *VS Code* dapat diakses di *GitHub*, yang telah membuatnya populer di kalangan pengembang aplikasi karena dapat berkontribusi dalam proses pengembangan yang berkelanjutan[13].



Gambar 2. 5 *Visual Studio Code*

2.2.6 Codeigniter

CodeIgniter adalah sebuah *framework* PHP (*Hypertext Preprocessor*) dengan mengadopsi pola *Model-View-Controller* (MVC). CI (*CodeIgniter*) juga dikenal sebagai *framework* yang mempunyai dokumentasi jelas dan lengkap. Kode sumber *CodeIgniter* dilengkapi dengan komentar yang membantu menjelaskan fungsi bagian program, sehingga memudahkan pemahaman. *CodeIgniter* menghasilkan kode yang bersih dan ramah terhadap mesin pencari (*Search Engine Friendly*, SEF). Framework ini juga mempermudah pengembang untuk membuat web dengan basis PHP (*Hypertext Preprocessor*), karena menyediakan kerangka yang sudah siap pakai sehingga mengurangi kebutuhan untuk menulis kode program dari awal[14].



Gambar 2. 6 Codeginitier

2.2.7 XAMPP

XAMPP adalah *software* sumber terbuka yang *support* berbagai OS (*Operating System*), yang menjadi gabungan dari lebih dari satu

program. XAMPP berfungsi menjadi server lokal yang terintegrasi, terdiri dari program utama seperti Apache HTTP (*Hyper Text Transfer Protokol*) Server, basis data MySQL, dan interpreter untuk bahasa program Perl dan PHP (*Hypertext Preprocessor*)[15].



Gambar 2. 7 XAMPP

2.2.8 RFID Tag

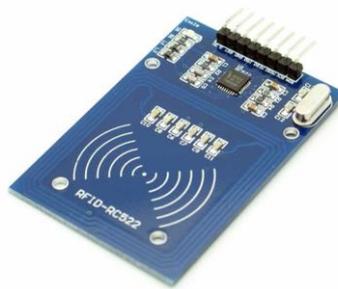
RFID (*Radio Frequency Identification*) Tag merupakan perangkat terdiri dari IC (*Integrated Circuit*) dan antena yang tersambung di dalamnya, dilengkapi dengan memori sehingga dapat menyimpan data. Secara umum, RFID (*Radio Frequency Identification*) Tag terbagi dalam dua jenis utama, yaitu tag pasif dan tag aktif[16].



Gambar 2. 8 RFID Tag

2.2.9 RFID Reader

RFID *Reader* merupakan modul yang berbasis MFRC522, dirancang untuk membaca kartu dengan cara yang sederhana dan dengan nominal yang terjangkau. Alat ini telah dilengkapi dengan semua perangkat yang diperlukan IC (*Integrated Circuit*) MFRC522 untuk beroperasi. Penggunaannya cukup mudah karena alat dapat dihubungkan langsung ke mikrokontroler menggunakan antarmuka SPI, dengan kebutuhan tegangan sekitar 3,3V. Desain RFID (*Radio Frequency Identification*) ini berguna untuk memahami karakteristik dari pembaca RFID (*Radio Frequency Identification*) yang mendukung berbagai jenis kartu, sehingga mempermudah para peneliti dalam merancang alat[17].



Gambar 2. 9 RFID *Reader*

2.2.10 Motor Servo

Motor servo merupakan aktuator yang menggunakan sistem kontrol untuk menentukan dan mempertahankan posisi sudut dari poros outputnya. Motor servo terdiri atas motor DC (*Direct Current*), rangkaian gear untuk mengurangi kecepatan poros dan menambah torsi,

rangkaiannya kontrol, dan potensiometer. Potensiometer ini berfungsi sebagai pengukur posisi sudut poros motor servo dengan mengubah resistansinya seiring dengan pergerakan motor. Dengan kombinasi komponen ini, motor servo dapat diatur secara presisi untuk berbagai aplikasi yang memerlukan kontrol posisi yang akurat[18].



Gambar 2. 10 Motor Servo

2.2.11 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) yaitu bahasa yang dipakai untuk mendokumentasikan, menentukan, dan membangun artefak dalam pembuatan perangkat lunak. UML (*Unified Modeling Language*) menyediakan *tool* untuk menciptakan gambaran visual model dari sistem. Sistem ini mencakup fungsi dan informasi, dan secara umum dipakai untuk memodelkan sistem komputer. UML (*Unified Modeling Language*) tidak hanya terbatas pada pemodelan, tetapi juga digunakan dalam berbagai bidang lain yang membutuhkan pemodelan[19].

1. Diagram *use case* adalah suatu metode pemodelan yang digunakan dalam menggambarkan aksi sistem informasi yang dibuat.

Diagram ini bekerja dengan cara menjelaskan komunikasi antara pengguna sistem dan sistem itu sendiri melalui narasi atau alur bagaimana penggunaan sistem tersebut terjadi[20].

Tabel 2. 1 Simbol *Usecase* Diagram

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Menentukan himpunan peran yang dimainkan pengguna saat berkomunikasi dengan <i>use case</i> berarti mendefinisikan peran atau karakteristik yang dimainkan oleh pengguna dalam konteks penggunaan tertentu dari suatu sistem atau aplikasi
2.		<i>Dependency</i>	Hubungan di mana perubahan pada elemen yang mandiri mempengaruhi elemen sebagai hubungan ketergantungan
3.		<i>Generalization</i>	Hubungan objek (<i>descendant</i>) berbagi sikap dan data struktur objek yang berada di atasnya.
4.		<i>Include</i>	Menentukan <i>use case</i> dengan jelas.
5.		<i>Extend</i>	Menentukan <i>use case</i> target meluaskan fungsionalitas dari <i>use case</i> sumber pada aksi tertentu.
6.		<i>Association</i>	Apa yang menyambungkan satu objek dari objek yang lain.
7.		<i>System</i>	Menentukan paket yang menggambarkan sistem dalam lingkup yang terbatas.
8.		<i>Use Case</i>	Deskripsi dari tindakan-tindakan yang dilakukan oleh sistem untuk menghasilkan data yang

No	Gambar	Nama	Keterangan
			dapat diukur bagi seorang aktor.
9.		<i>Collaboration</i>	Komunikasi antara aturan dan elemen yang berinteraksi bersama untuk menghasilkan perilaku yang lebih luas daripada sekadar jumlah dari aturan dan elemen-elemen tersebut.
10.		<i>Note</i>	Fisik Elemen yang ada ketika aplikasi berjalan dan merepresentasikan sebuah sumber daya komputasi.

2. *Activity Diagram* adalah representasi diagramatis dari aliran kegiatan atau alur kerja dalam sebuah sistem yang dieksekusi [21].

Tabel 2. 2 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Menunjukkan setiap kelas antarmuka berkomunikasi satu dengan yang lain.
2.		<i>Action</i>	Status yang merefleksikan pelaksanaan sebuah aksi.
3.		<i>Initial Node</i>	Bagaimana objek diciptakan atau diinisiasi.
4.		<i>Final Node</i>	Bagaimana objek diciptakan dan dihapus.
5.		<i>Fork Node</i>	Suatu aliran suatu titik berubah menjadi beberapa.
6.		<i>Decision</i>	Keputusan untuk membuat pilihan.

No	Gambar	Nama	Keterangan
7.		<i>Fork/Join</i>	Menunjukkan aktivitas dengan simultan atau untuk menggabungkan dua kegiatan lebih menjadi satu.
8.		<i>Rake</i>	Menyatakan proses dekomposisi.
9.		<i>Time</i>	Indikator waktu.
10.		<i>Send</i>	Indikator pengiriman.

3. *Class Diagram* adalah diagram yang mampu memetakan struktur suatu sistem dengan jelas melalui pemodelan operasi, atribut, kelas, dan hubungan objek[22].

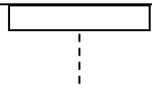
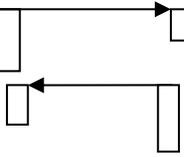
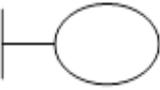
Tabel 2. 3 Simbol *Class Diagram*

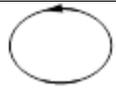
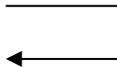
No	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendant</i>) mewarisi tindakan dan struktur data dari induk objek.
2.		<i>Nary Association</i>	Bagian yang menghindari keterkaitan dari dua objek.
3.		<i>Class</i>	Sekelompok objek yang memiliki atribut dan fungsi sama.
4.		<i>Collaboration</i>	Deskripsi tindakan yang dilakukan oleh sistem untuk

No	Gambar	Nama	Keterangan
			menghasilkan nilai terukur untuk seorang aktor.
5.		<i>Dependency</i>	Tindakan yang sebenarnya dilakukan dari sebuah objek.
6.		<i>Dependency</i>	Perubahan hubungan pada elemen independen yang mempengaruhi elemen lain yang bergantung padanya.
7.		<i>Association</i>	Menghubungkan satu objek dari objek yang lain.

4. *Sequence diagram* memvisualisasikan interaksi antara objek-objek dalam (termasuk pengguna, tampilan, dan sebagainya) melalui pesan-pesan yang ditampilkan seiring waktu. *Sequence diagram* memiliki dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terlibat)[23].

Tabel 2. 4 Simbol Sequence Diagram

No	Gambar	Nama	Keterangan
1.		<i>Life Line</i>	Entitas objek yang berinteraksi antar muka.
2.		<i>Message</i>	Spesifikasi komunikasi objek yang mencakup informasi aktivitas terjadi.
3		<i>Actor</i>	Menggambarkan individu yang sedang berkomunikasi dengan sistem.
4		<i>Boundary Class</i>	Menggambarkan representasi visual dari formulir.

No	Gambar	Nama	Keterangan
5		<i>Entity Class</i>	Menggambarkan interaksi aktivitas yang dilakukan.
6.		<i>Control Class</i>	Menggambarkan koneksi antara <i>Boundary</i> dengan tabel.
7		<i>Activation</i>	Sebagai suatu objek yang menjalankan suatu tindakan.
8	Message 	<i>Message</i>	Menggambarkan interaksi antara objek dan objek yang lain.
9		<i>Self Message</i>	Menggambarkan komunikasi kembali ke dalam objek sendiri.