

BAB II

KAJIAN PUSTAKA

A. Telaah Pustaka

Telaah Pustaka sebagai bahan acuan dan referensi dalam mengimplementasikan sistem monitoring perawatan berkala pergantian oli mesin sepeda motor dengan jarak tempuh, maka penelitian menggunakan beberapa penelitian sejenis yang berkaitan dengan variabel penelitian yang relevan dengan hasil penelitian lain. Berikut beberapa penelitian terdahulu yang berkaitan dengan penelitian ini yaitu:

Rama Ramanda, Oktef Brilliant Kharisma, Alex Wenda, dan Abdilah (2021) dengan judul Sistem Pemantauan Kelayakan Pelumas Oli Pada Kendaraan Sepeda Motor Dengan Memanfaatkan Teknologi *Internet Of Things* (IOT). Tujuan penelitian ini adalah akan dirancang suatu sistem pemantauan pergantian oli sepeda motor berdasarkan jarak tempuh dan waktu pemakaian oli yang berbasis *IoT*. Dimana penggunaan sepeda motor digunakan untuk mensimulasikan perputaran roda, sensor proximity sebagai pendeteksi putaran roda, serta mikrokontroler *ESP32* sebagai pengendali utama, buzzer untuk memperingati pengguna, relay digunakan untuk mematikan mesin sepeda motor, dan *GSM* pengirim data. Penelitian ini diperoleh hasil yang diharapkan, Ketika jarak tempuh mencapai jarak 2000 km maka sistem akan memperingati pengguna sepeda motor dan Ketika pengguna mengembalikan peringatan tersebut dan jarak mencapai 2100 km maka sistem mematikan kelistrikan pada sepeda motor. Namun jika jarak tidak tercapai tetapi pemakaian oli telah mencapai 60 hari maka sistem memperingati pengguna sepeda motor.

Ketut Abimanyu, Nina Lestari, Muhammad Anton Fauzi, dan Aji Nurcahya (2020) dengan judul Perancangan Sistem Monitoring Penggantian Oli Pada Sepeda Motor Berdasarkan Jarak Tempuh. Tujuan dari penelitian ini adalah bagaimana cara membandingkan jarak tempuh yang dihitung oleh sistem dengan speedometer digital pada kendaraan. Selain itu sistem harus bisa mendeteksi kapan kendaraan harus melakukan penggantian oli setelah mencapai jarak tempuh tertentu. Pada penelitian ini, sensor yang digunakan adalah sensor proximity yang berkerja secara induktif. Digunakan pula *Arduino Uno* sebagai pengolah data, dimana hasil pengolahan data akan ditampilkan dalam bentuk pesan peringatan ganti oli yang

akan menggunakan LED inductor dan ditampilkan di LCD. Berdasarkan tes uji, system dapat menampilkan peringatan penggantian oli pada jarak tertentu dengan hasil akurasi sistem sebesar 10 % jarak tempuh 10 kilo meter (km).

Holis Muchlis Sugianto dan Asti herlina (2020) dengan judul Perancangan Aplikasi Reminder dan Minotoring Jadwal Servis Sepeda Motor Berbasis Android. Tujuan penelitian ini adalah diusulkan sebuah aplikasi pengingat dan penjadwalan servis agar pengguna sepeda motor dapat membuat pengingat yang dapat digunakan untuk penjadwalan servis sehingga dapat menghindari keterlambatan dalam melakukan servis. Pada penelitian kali ini metode yang digunakan adalah metode kualitatif, sedangkan perancangan sistem pada aplikasi ini menggunakan bahasa pemrograman Java, XML, dan SQLite sebagai database. Dengan adanya aplikasi reminder dan monitoring jadwal servis sepeda motor berbasis android ini diharapkan dapat membantu para pengguna sepeda motor dalam membuat pengingat servis secara rutin dan mengetahui kapan sebaiknya sepeda motor dapat diservis.

Nurus Sholeh, Koko Joni, dan Miftachul Ulum (2020) dengan judul system monitoring kondisi kendaraan motor injeksi berbasis mikrokontroler. Tujuan penelitian ini adalah maka dibuatlah sebuah sistem berbasis mikrokontroler yang memberikan peringatan kepada pemilik kendaraan sepeda motor vixion. Mikrokontroler merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu chip IC, sehingga sering disebut single chip microcomputer. Dimana sistem ini mengutamakan ketepatan waktu dalam pergantian oli, air radiator dan indikator kerusakan dalam sepeda motor. Penyampaian informasi oli menggunakan buzzer. Radiator dan indikator kerusakan pada sepeda motor di lakukan melalui sms gateway ke smart phone atau android. Untuk itu diperlukan data-data yang di inputkan kedalam suatu system untuk dibaca oleh alat tersebut. Alat-alat yang digunakan untuk pemantauan kondisi motor antara lain sensor kecepatan, arduino, AT MEGA 8535, Buzzer dan sms gateway.

Muchsin Attubel, Diky Siswanto, dan Mohammad Muchsin (2019) dengan judul Sistem Pemantauan dan Peringatan Waktu Perawatan Kendaraan Berbasis *Internet Of Thinks* (IOT). Tujuan dari penelitian ini adalah untuk merancang sistem monitoring perawatan kendaraan bermotor. Dari pengujian diperoleh battery aki

kendaraan mampu mencatu daya hardware selama 2,33 hari. Selain itu diperoleh jarak antara titik acuan dengan lokasi terdeteksi memiliki selisih rata-rata sebesar 1,75m. Juga diperoleh tingkat keakuratan *global positioning system* (GPS) pada kendaraan (perangkat yang dirancang) lebih tinggi jika dibandingkan dengan GPS smartphone. Berdasarkan pengujian keseluruhan menunjukkan rata-rata delay pengiriman data dari kendaraan hingga sampai ke user adalah 1,36 detik. Secara keseluruhan perangkat yang dibuat berfungsi sesuai dengan hasil rancangan.

Ratih Ayuninghemi dan Mochammad Fakhri Abdillah (2018) dengan judul Simulasi Alat Notifikasi Servis Ringan Sepeda Motor Otomatis Berbasis Arduino Nano Terintegrasi Dengan Android. Tujuan penelitian ini adalah menghasilkan alat hitung digital (speedometer android) yang membaca data dari arduino nano sebagai mikrokontroler dan terhubung secara wireless dengan Bluetooth, alat ini diuji dengan 2 sepeda motor yang berbeda tipe jarak pembacaan jarum pada speedometernya, percobaan pertama menggunakan sepeda motor mio sporty yang mendapatkan hasil notifikasi muncul pada saat mencapai batas kilometer tercapai tetapi terdapat perbedaan hasil perhitungan jarak antara speedometer dan speedometer android, percobaan kedua menggunakan sepeda motor mio jet fi yang mendapatkan hasil notifikasi muncul pada saat batas kilometer telah tercapai dan hasil cukup akurat. Dan percobaan dengan jarak bulan waktu servis yang mendapatkan hasil akurat.

Renda Arya Santosa, Diah Riskiwati, dan Zamah Sari (2017) dengan judul rancang Bangun Sistem Informasi Service Oli Sepeda Motor Dengan Odometer Berbasis Located Based Service. Penggunaan sensor magnet Reed Switch untuk menghitung putaran roda sepeda motor, kemudian Mikrokontroler digunakan sebagai pengolah data putaran roda menjadi jarak tempuh. Apabila perhitungan jarak tempuh telah mencapai batas untuk servis oli, maka akan diberikan suatu peringatan berupa suara menggunakan buzzer. Sistem ini juga dilengkapi dengan button reset untuk melakukan reset perhitungan jarak tempuh dan sd card yang digunakan untuk menyimpan data jarak tempuh. Aplikasi Location Based Service dapat menentukan posisi pengguna maupun bengkel-bengkel resmi Honda yang telah terintegrasi. Dengan kombinasi ini, aplikasi Location Based Service bengkel akan mencari rute untuk menghubungkan posisi pengguna dengan bengkel motor

serta memanfaatkan servis di google yang sudah disertakan layer traffic sehingga pengguna dapat memilih bengkel yang sesuai dengan keinginannya. Dari hasil aplikasi Location Based Service bengkel yang telah dibuat, aplikasi dapat mendaftarkan servis sepeda motor ke bengkel-bengkel yang terintegrasi. Aplikasi tersebut mampu melakukan proses pencarian lokasi bengkel terdekat di sekitar pengguna dan menunjukkan rute dari lokasi pengguna ke bengkel tujuan menggunakan API google map.

Berdasarkan beberapa uraian diatas, maka usulan yang diajukan pada penelitian ini adalah Sistem Monitoring Perawatan Berkala Penggantian Oli Mesin Sepeda Motor dengan Jarak Tempuh Menggunakan NodeMCU ESP8266. Sehingga dapat mempermudah pengguna sepeda motor dalam melakukan pengingat waktu pergantian oli mesin dan suku cadang sepeda motor berdasarkan jarak tempuh, selain sebagai pengingat sistem ini juga bisa menjaga mesin tetap prima dan tidak aus disaat pergantian oli mesin melebihi batas waktu pergantian. Pada penelitian (Ketut Abimanyu, Nina Lestari, Mohammad Anton Fauzi, dan Adi Nurcahya, 2020) terdapat kesamaan yaitu terkait pergantian oli mesin sepeda motor menggunakan jarak tempuh dengan sensor proximity, sedangkan pada penelitian ini dapat menginformasikan pergantian suku cadang sepeda motor dengan offset jarak tempuh. Pada penelitian yang dilakukan oleh Holis Muchlis Sugianto dan Asti Herlina (2020) yaitu berbeda dalam penggunaan database server.

B. Landasan Teori

1. Profil Bengkel Lancar Motor 45

Bengkel Lancar Motor 45 adalah bengkel independen yang bergerak dalam bidang jasa dan perbaikan. Jangkauan produktifitas Bengkel Lancar Motor 45 meliputi service, tune-up, dan restorasi serta menyediakan suku cadang sepeda motor. Bengkel Lancar Motor 45 berkomitmen untuk bergerak maju dengan kuat sejalan dengan tren teknologi industry pasar yaitu pemantauan perawatan berkala sepeda motor berbasis aplikasi.

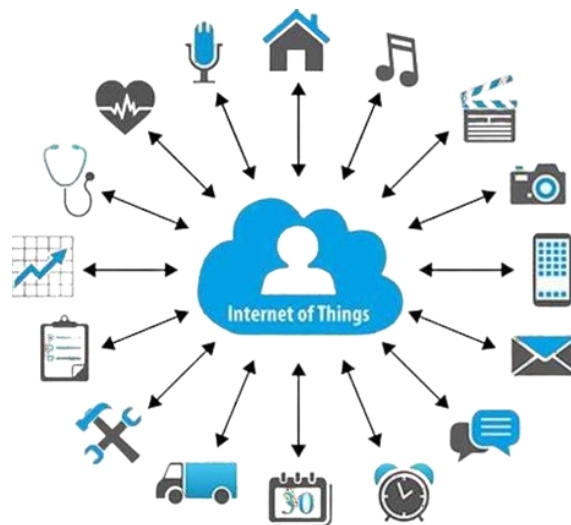
Sejarah Bengkel Lancar Motor 45 berdiri sejak tahun 2012, yaitu didirikan oleh anak keempat dari lima bersaudara. Berlokasi cukup strategis di Jalan Jati Padang Raya Pintu Timur Ragunan, Jakarta Selatan. Kemudian pada tahun 2021 bulan

agustus bercabang didesa Tunjung Kecamatan Jatilawang Kabupaten banyumas dengan nama Bengkel Lancar Motor 45 cabang Jakarta Selatan. Dimana melayani berbagai merk kendaraan sepeda motor seperti Honda, Yamaha, Suzuki, dan Kawasaki. Menajement pengelolaan dan operasionalnya dilaksanakan secara professional dimana pemiliknya Bapak Noto mempunyai pengalaman di bisnis otomotif dan distribusi suku cadang sepeda motor meliputi wilayah Jakarta dan Banyumas.

Bengkel Lancar Motor 45 mempunyai slogan “mengangkat ESDM generasi muda melalui otomotif” yaitu merupakan program pelatihan untuk generasi muda yang minim Pendidikan dan minim Pengetahuan di bidang otomotif. Dimana sebagai motivasi menjalankan visi dan misi perusahaan.

2. Internet Of Thinks (IoT)

IoT atau yang disebut *Internet Of Thinks* merupakan salah satu teknologi yang memanfaatkan sambungan internet yang selalu terkoneksi setiap saat (Hasiholan, Primananda, and Amron 2018). *Internet Of Thinks* memungkinkan benda-benda disekitar kita terhubung dengan jaringan internet yang dapat dimonitor dan dikendalikan dari jarak jauh. Perkembangan teknologi *Internet Of Thinks* hingga kini sudah banyak dikembangkan dan di aplikasikan dalam kehidupan. Cara kerja *Internet Of Thinks* adalah setiap benda yang sudah terpasang sensor atau modul *Internet Of Thinks* itu sendiri adalah untuk memudahkan *monitoring* dan *controlling* suatu benda pada kehidupan sehari-hari. Selain itu informasi yang didapat bisa setiap watu pada *Internet Of Thinks*.



Gambar.1. Ilustrasi *internet Of Thinks*

3. Node MCU ESP 8266

Node MCU ESP 8266 adalah sebuah platform IoT yang bersifat *Open Source*. ESP8266 dari seri ESP besutan *Espressif System*, juga firmware yang digunakan merupakan bahasa pemrograman *Scripting Lua*. Istilah NodeMCU secara default sebenarnya mengacu pada firmware yang digunakan daripada perangkat keras development kit, dan NodeMCU juga bisa diartikan sebagai board arduinonya ESP8266. Selain dengan bahasa lua NodeMCU juga support dengan software Arduino IDE dengan melakukan sedikit perubahan pada board manager di dalam software Arduino IDE yaitu dengan menambahkan URL untuk mengunduh board khusus NodeMCU pada board manger (Mohamad Yusuf Efendi and Joni Eka Chandra 2019).



Gambar.2. Node MCU ESP 8266

4. Sensor Proximity

Sensor Proximity berkerja berdasarkan jarak objek berdasarkan sensor, Ketika ada objek logam yang mendekat kepadanya dengan jarak yang sangat dekat, sensor akan berkerja dan menghubungkan kontaknya. Kemudian melalui kabel yang tersedia bisa dihubungkan ke perangkat lainnya seperti lampu indicator, relay, dan lain-lainnya. Pada saat sensor ini sedang berkerja atau mendeteksi adanya logam (besi) maka akan ditandai dengan lampu kecil berwarna merah atau hijau yang ada

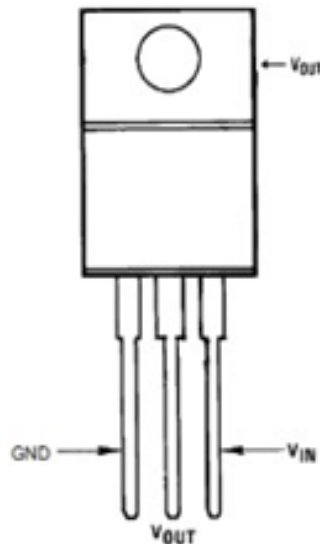
dibagian atas sensor, sehingga memudahkan kita dalam memonitor kerja sensor atau Ketika melakukan *preventive maintenance* (Rahmat Farhan, Muhaimin, and, Maimun 2019).



Gambar.3. Sensor Proximity

5. IC Regulator LM7805

IC Regulator yang digunakan adalah menjadi 5 volt untuk memberikan supply LM7805. IC Regulator digunakan sebagai tegangan pada *NodeMCU ESP8266* untuk penstabil tegangan 12 volt dari *accu* motor (Texas Instrumen, “LM340, LM340A, and LM7805” 2017).



Gambar.4. IC Regulator LM7805

6. Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loudspeaker. Buzzer terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi electromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap Gerakan kumparan akan menggerakkan secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer bisa digunakan sebagai indicator bahwa proses telah selesai atau menjadi suatu kesalahan pada sebuah alat.



Gambar.5. *Buzzer alarm*

7. Arduino Ide

IDE Arduino merupakan software yang menyerupai bahasa C dan ditulis dengan menggunakan java. *IDE Arduino* terdiri dari editor program, window yang memungkinkan pengguna membuat dan mengedit program dalam bahasa *Processing*. *Compiler* pada arduino adalah sebuah modul yang mengubah program bahasa *processing* menjadi biner. Yang dipahami oleh mikrokontroler adalah kode biner (A. Antu, Abdussamad, and Z. Nasibu 2020).



Gambar.6. *Arduino IDE*

8. Android Studio

Android Studio adalah sebuah IDE yang bisa digunakan untuk pengembangan aplikasi android, dan dikembangkan oleh google. Android Studio merupakan pengembangan dari Eclipse IDE Java populer, yaitu IntelliJ IDEA. Android Studio direncanakan untuk menggantikan Eclipse kedepannya dengan IDE resmi untuk pengembangan aplikasi android. Sebagai pengembangan dari Eclipse Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan ADT, Android Studio menggunakan gradle sebagai *build environment* (Makiolor, Sinsuw, and B.N. Najoan 2017).



Gambar.7. *Android Studio*

9. Firebase

Firebase adalah layanan Google yang berguna untuk membantu para developer untuk mengembangkan aplikasi. Firebase atau *Backend as a Service* merupakan cara developer yang fokus mengembangkan aplikasi tanpa perlu effort

besar. Sejak ditemukan oleh Andrew Lee dan James Tamplin tahun 2011, Firebase memberikan produk Realtime Database yang berguna untuk menyimpan lebih banyak data dan sinkronasi ke banyak pengguna. Pada tahun 2014 lalu, pihak Google mengakuisisi layanan pengembang aplikasi ini (Andy Nugroho 2021).



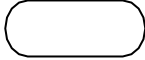
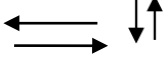
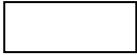
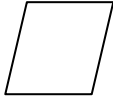
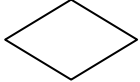


Gambar.8. Firebase

10. *Unified Modeling Language (UML)*

UML (Unified Modeling Language) adalah bahasa untuk membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi, atau perangkat lunak) dari sistem perangkat lunak. Selain itu, *UML* adalah Bahasa pemodelan yang menggunakan konsep orientasi *object*. *UML* menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. *UML* tidak hanya digunakan dalam pemodelan perangkat lunak, namun hamper dalam semua bidang yang membutuhkan pemodelan (Ipo Novianto 2017).

11. Flowchart

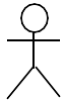
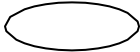

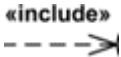
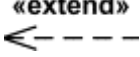

Flowchart adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma (Ipo Novianto 2017). Simbol *Flowchart* dapat dilihat pada tabel berikut.

No	Simbol	Nama Simbol dan Keterangan
1.		Simbol Terminal yaitu untuk menyatakan permulaan atau akhir suatu program.
2.		Simbol Arus / Flow yaitu untuk menyatakan jalannya arus suatu proses (arah aliran program).
3.		Simbol Proses yaitu untuk menyatakan proses perhitungan / proses pengolahan data.
4.		Simbol Input - Output yaitu untuk memasukan data (proses <i>input</i>) maupun menunjukan hasil (<i>output</i> data) dari suatu proses.
5.		Simbol Decision / Logika yaitu untuk menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya / tidak.
6.		Simbol Document yaitu untuk mencetak laporan ke printer
7.		Simbol Manual Operation adalah simbol berfungsi untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer.

12. Use Case diagram




Use case adalah abstraksi dari interaksi antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan

terlihat di mata *user*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client* (Ipo Novianto 2017).

No	Simbol	Nama Simbol dan Keterangan
1.		Actor adalah menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		Use Case menjelaskan urutan kegiatan yang dilakukan <i>actor</i> dan sistem untuk mencapai suatu tujuan tertentu.
3.		Association penghubung antara <i>Actor</i> dan <i>Use Case</i>
4.		Include yaitu perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.
5.		Extend yaitu perilaku suatu <i>use case</i> memperluas perilaku <i>use case</i> lain. <i>Use case</i> boleh dilakukan boleh tidak (optional)
6.		Sistem Boundary yaitu batasan dari sebuah sistem.

13. Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima oleh *objek*. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *diagram sequence* juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut adalah simbol-simbol yang ada pada *diagram sequence* (Ipo Novianto 2017).

No	Simbol	Nama dan Keterangan
1.		<i>Asosiasi</i> adalah hubunga antar kelas.
2.		<i>Generalization</i> adalah relasi antar kelas dengan makna generalisasi – spesialisasi (umum-khusus).
3.		<i>Composition</i> adalah bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> dibuat. Misal kelas <i>whole</i> dihapus, maka kelas yg menjadi part ikut musnah.




14. Class Diagram

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita gunakan. *Class diagram* memberi kita gambaran tentang perangkat lunak dan relasi-relasi yang ada didalamnya. Menurut (Whitten L. Jeffery et al 2004:432) *Class Diagram* adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu *class diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk *UML* (Ipo Novianto 2017).

Atribut dan operation dapat memliki salah satu sifat berikut :

- Private hanya bisa dipanggil dari dalam kelas iu sendiri. Atribut diawali “ - ”.
- Protected hanya dapat dipanggil oleh class yang bersangkutan dan class turunannya. Atribut diawali dengan tanda “ # ”.
- Public dapat dipanggil dari semua objek atribut diawali tanda “ + ”.

Ada beberapa symbol relationships antar class yang digunakan pada diagram class. Berikut adalah beberapa symbol-simbol yang ada pada class diagram dapat dilihat pada tabel.98.

No	Simbol	Nama dan Keterangan
1.		<i>Asosiasi</i> adalah hubungan antar kelas.
2.		<i>Generalization</i> adalah relasi antar kelas dengan makna generalisasi – spesialisasi (umum-khusus).
3.		<i>Composition</i> adalah bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi <i>whole</i> dibuat. Misal kelas <i>whole</i> dihapus, maka kelas yg menjadi part ikut musnah.

Multiplicity adalah jumlah banyaknya objek sebuah kelas yang berelasi dengan sebuah objek *dari* kelas lain yang berasosiasi dengan kelas tersebut (Ipo Novianto 2017). Macam-macam *multiplicity* disajikan dalam Tabel.

Nilai Kardinalitas	Arti
0..1	Nol atau satu
1	Hanya satu
0..*	Nol atau lebih
1..*	Satu atau lebih
N	Hanya n (dengan $n > 1$)
0..n	Nol sampai n (dengan $n > 1$)
1..n	Satu sampai n (dengan $n > 1$)

15. Prototype

Prototype merupakan salah satu dari implementasi sebuah desain produk yang akan dibangun. *Prototype* adalah sebuah bukti fisik atau konsep dari sebuah konsep perancangan. *Prototype* dapat membantu untuk menentukan apakah pengembang aplikasi telah berada pada jalur yang benar selama proses pengembangan. Pelatih dan pengembang profesional menggunakan teknik *prototyping* untuk menghasilkan umpan balik dari pengguna akhir selama proses perancangan sebelum akhirnya mulai membuat code aplikasi. Mendapatkan umpan balik yang kritis akan membantu dalam menghasilkan desain akhir yang pada akhirnya berdampak besar pada pengguna (Martono 2018).