**Sistem Pengelolaan Parkir Otomatis Menggunakan Metode Unified Modelling Language (UML)**

## TUGAS PRAKTIKUM

Disusun oleh:

|  |  |
| --- | --- |
| **Rani Wahyu Aprilia** | **3311811029** |
| **Reynaldi Sihombing** | **3311811037** |
| **Dwi Aji** | **3311811042** |

Disusun untuk memenuhi salah satu syarat kelulusan matakuliah IF312 Rekayasa Perangkat Lunak II



## PROGRAM STUDI TEKNIK INFORMATIKA

**JURUSAN TEKNIK INFORMATIKA POLITEKNIK NEGERI BATAM**

**BATAM 2019**

## HALAMAN PENGESAHAN

**Sistem Pengelolaan Parkir Otomatis   
Menggunakan Metode Unified Modelling Language (UML)**

## Disusun oleh:

|  |  |
| --- | --- |
| **Rani Wahyu Aprilia** | **3311811029** |
| **Reynaldi Sihombing** | **3311811037** |
| **Dwi Aji** | **3311811042** |

Batam, 15 Oktober 2019

Disetujui dan disahkan oleh: Dosen pengajar,

## Rina Yulius

**NIK/NIP.**

## HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311811029

Nama : Rani Wahyu Aprilia

adalah mahasiswa Program Studi Teknik Informatika Politeknik Negeri Batam menyatakan bahwa Tugas Praktikum dengan judul:

**Sistem Pengelolaan Parkir Otomatis   
Menggunakan Metode Unified Modelling Language (UML)**

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya oranglain
2. tidak melakukan pemalsuandata
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijinpemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Praktikumini.

Batam, 15 Oktober 2019

## Rani Wahyu Aprilia NIM. 3311811029

## HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311811037

Nama : Reynaldi Sihombing

adalah mahasiswa Program Studi Teknik Informatika Politeknik Negeri Batam menyatakan bahwa Tugas Praktikum dengan judul:

**Sistem Pengelolaan Parkir Otomatis   
Menggunakan Metode Unified Modelling Language (UML)**

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya oranglain
2. tidak melakukan pemalsuandata
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijinpemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Praktikumini.

Batam, 15 Oktober 2019

## Reynaldi Sihombing NIM.3311811037

## HALAMAN PERNYATAAN

Dengan ini, saya:

NIM : 3311811042

Nama : Dwi Aji

adalah mahasiswa Program Studi Teknik Informatika Politeknik Negeri Batam menyatakan bahwa Tugas Praktikum dengan judul:

**Sistem Pengelolaan Parkir Otomatis   
Menggunakan Metode Unified Modelling Language (UML)**

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya oranglain
2. tidak melakukan pemalsuandata
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijinpemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Praktikumini.

Batam, 15 Oktober 2019

## Dwi Aji NIM. 3311811042

**ABSTRACT**

**Sistem Pengelolaan Parkir Otomatis   
Menggunakan Metode Unified Modelling Language (UML)**

Sistem Parkir tidak lagi harus dilakukan secara manual. Seiring kemajuan pengetahuan informasi dan teknologi, parkir dapat dilakukan secara otomatis. Parkir otomatis dapat dijumpai pada pusat perbelanjaan, bandara, rumah sakit dan kampus. Dengan adanya sistem pengelolaan parkir otomatis pada kampus, perusahaan bisa meningkatkan pendapatan dan keuntungan, serta bisa memangkas biaya-biaya operasional lainnya. mahasiswa juga bisa mendapatkan efisiensi waktu dan lebih fleksibel dalam memarkirkan kendaraan dan bertransaksi. Parkir otomatis ini memberikan rasa aman secara fisik karena penjaga parkir tidak perlu mendatangi menghitung biaya parkir, menghitung lama waktu kendaraan diparkir, serta menulis nomor kendaraan yang akan di parkir secara manual dan ini memungkinkan mahasiswa dapat bertransaksi secara aman sebab apabila dilakukan secara manual tingkat kesalahannya akan semakin besar. Parkir otomatis pada gilirannya akan merangsang orang-orang gatek (gagap teknologi) untuk mempelajari teknologi sistem informasi ini demi kepentingan mereka sendiri, sehingga hal ini dapat meningkatkan kualitas sumber daya manusia. Dalam melakukan penelitian penulis menggunakan metode penelitian yaitu, studi kepustakaan. Pembuatan Sistem Informasi Pengelolaan Parkir Otomatis menggunakan bahasa pemodelan UML (Unified Modelling Language).

Kata Kunci : Parkir Otomatis, sistem parkir, UML.

## DAFTAR ISI

1. HalamanJudul 1
2. [HalamanPengesahan 2](#_bookmark0)
3. [HalamanPernyataan 3](#_TOC_250015)

[HalamanPernyataan 4](#_TOC_250014)

[HalamanPernyataan 5](#_TOC_250013)

1. Abstrak 6
2. [DaftarIsi… 7](#_TOC_250012)
3. Sejarah UML 8

[Bab 1 Pendahuluan 1](#_bookmark1)1

* 1. [Latar Belakang 11](#_TOC_250010)
  2. [Maksud dan Tujuan 12](#_TOC_250009)
  3. [Ruang Lingkup 12](#_TOC_250008)

Bab 2 Landasan Teori

[2.1 Pengertian UML 14](#_TOC_250006)

* 1. Diagram-Diagram UML…………………………………..................................

Bab 3 Sistem Berjalan 15

* 1. [Activity Diagram… 15](#_TOC_250005)

Bab 4 Perancangan Sistem Usualan dengan UML 32

* 1. Diagram Sistem

4.1.1 Use Case Diagram 32

4.1.2 Activity Diagram 32

4.1.3 Sequence Diagram....................................................................................

4.1.4 Class Diagram…………………………………………………………..

* 1. Desain Data……………………………………………………………..

1. ERD………………………………………………………………….
2. Spesifikasi File………………………………………………………..
   1. Spesifikasi Hardware & Software………………………………………..
   2. User Interface……………………………………………………………

Bab 5

5.1 Kesimpulan & Saran

5.2 Referensi

**VI. Sejarah UML**

Sejarah UML sendiri cukup panjang. Tahun 1950-an saat keterbatasan hardware, media penyimpanan dan software pemrograman, muncul metode perancangan sistem yang berbasis proses. Muncul diagram-diagram terkenal seperti Data Flow Diagram (DFD). Inti dari diagram ini adalah entitas apa dan melakukan proses apa dengan metode yang sangat terkenal SDLC: System Development Life Cycle. Tahun 1976, Chen menemukan Entity Relationship Diagram (ERD) yang berguna dalam memodelkan database dari suatu proses. Dimulailah era metode perancangan sistem berbasis DATA. Muncul istilah terkenal: Relational Database Management System (RDBMS). Metode perancangan ini berusaha menutupi kelemahan metode perancangan berbasis proses. Perlu diketahui bahwa proses sangat cepat berubah dibandingkan data.

Sampai era tahun 1990, Tahun 90-an, diiringi membanjirnya software berorientasi object, bahkan hingga ke database seperti Oracle, SQL Server, dan lain-lain sudah menganut OR-DMBS (Object Relational – DBMS).  Seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: metodologi booch [1], metodologi coad [2], metodologi OOSE [3], metodologi OMT [4], metodologi shlaer-mellor [5], metodologi wirfs-brock [6], dsb. Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group. Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. Hingga saat ini UML sudah versi 2.2.

## Bab I

## Pendahuluan

## Latar Belakang

## Tempat parkir merupakan kebutuhan mutlak bagi semua orang. Sistem Parkir tidak lagi harus dilakukan secara manual. Seiring kemajuan pengetahuan informasi dan teknologi, parkir dapat dilakukan secara otomatis. Parkir otomatis dapat dijumpai pada pusat perbelanjaan, bandara, rumah sakit dan kampus. Dengan adanya sistem pengelolaan parkir otomatis pada kampus, pihak kampus bisa meningkatkan pendapatan dan keuntungan, serta bisa memangkas biaya-biaya operasional lainnya. mahasiswa juga bisa mendapatkan efisiensi waktu dan lebih fleksibel dalam memarkirkan kendaraan dan bertransaksi. Parkir otomatis menggunakan system *capture* ketika pemilik kendaraan ingin masuk. Sehingga parkir otomatis ini memberikan rasa aman secara fisik karena penjaga parkir tidak perlu mendatangi menghitung biaya parkir, menghitung lama waktu kendaraan diparkir, serta menulis nomor kendaraan yang akan di parkir secara manual dan ini memungkinkan mahasiswa dapat bertransaksi secara aman sebab apabila dilakukan secara manual tingkat kesalahannya akan semakin besar. Jenis pengguna parkir pun ada 2 yaitu mahasiswa dan staf/dosen. Untuk mahasiswa menggunakan karcis dan untuk staf/dosen menggunakan system *tapping* KTM/ID Card.

## Maksud dan Tujuan Tujuan penulisan melakukan penelitian adalah sebagai berikut:

## Agar pembaca dapat mengetahui Sistem Informasi Pengelolaan Parkir Otomatis pada Kampus

## Agar pembaca dapat memahami UML*(Unified Modeling Language)* dari parkir otomatis

## Dalam laporan ini penulis membahas tentang Sistem Informasi Pengelolaan Parkir Otomatis pada Kampus. Disini penulis menggunakan Ms.Visio untuk membuat Database,Use Case Diagram, Activity Diagram, Sequence Diagram, Class Diagram.

## BAB II Landasan Teori

## 2.1 Pengertian UML

**UML** adalah bahasa untuk menspesifikasi,memvisualisasi, 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,seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi *object*.UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corps*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai prespetktif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

## 2.2 Diagram-Diagram UML

**Use Case Diagram**Menggambarkan sejumlah external actors dan hubungannya ke use case yang diberikan oleh sistem. Use case adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari use case symbol namun dapat juga dilakukan dalam activity diagrams. Use case digambarkan hanya yang dilihat dari luar oleh actor (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem.

**Activity Diagram** Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

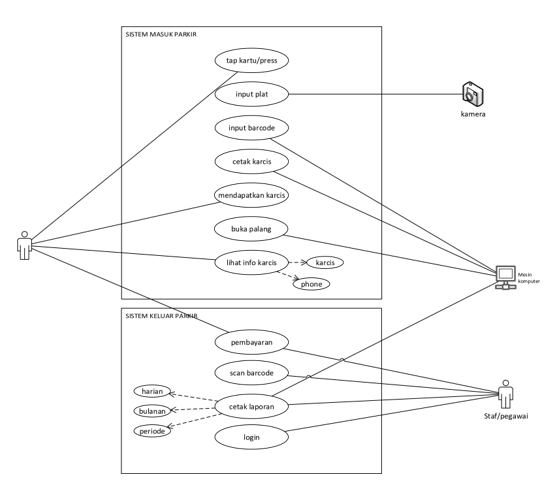
**Sequence Diagram**Menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

**Class Diagram** Menggambarkan struktur statis class di dalam sistem. Class merepresentasikan sesuatu yang ditangani oleh sistem. Class dapat berhubungan dengan yang lain melalui berbagai cara: associated (terhubung satu sama lain), dependent (satu class tergantung/menggunakan class yang lain), specialed (satu class merupakan spesialisasi dari class lainnya), atau package (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa class diagram.

**BAB III  
Sistem Berjalan**

3.1 Activity Diagram

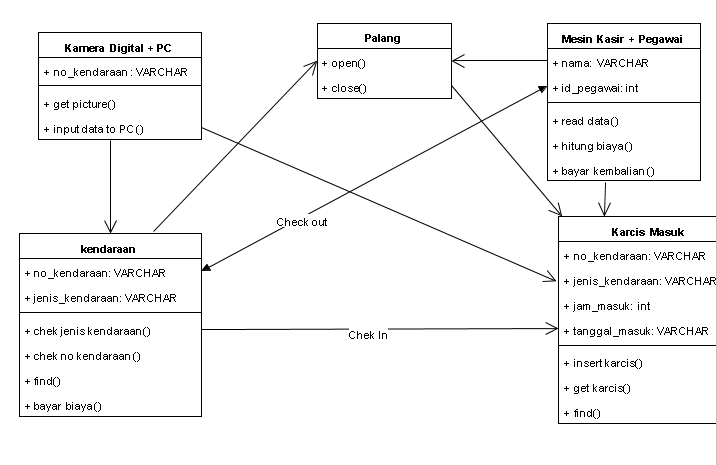
**BAB IV  
Perancangan Sistem Usualan dengan UML**

4.1 Diagram Sistem  
 4.1.1 Use Case Diagram  
 

4.1.2 Activity Diagram

4.1.3 Sequence Diagram

4.1.4 Class Diagram



1. Use Case Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama | Simbol | Fungsi |
| 1 | Actor |  | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case* |
| 2 | Dependecy |  | hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan memperngaruhi elemen yang bergantung padanya elemen yang tidak mandiri(*independet*) |
| 3 | Generalization |  | hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor). |
| 4 | Include |  | menspesifikasikan bahwa use case sumber secara eksplisit. |
| 5 | Association |  | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 | System |  | menspesifikasikan paket yang menampilkan sistem secara terbatas |
| 7 | Use Case |  | deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 8 | Collaboration |  | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi) |
| 9 | Note |  | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |

1. Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama | Simbol | Fungsi |
| 1 | Activity |  | memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 | Action |  | state dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 | Initial Node |  | bagaimana objek dibentuk atau diawali |
| 4 | Final Node |  | bagaimana objek dibentuk dan dihancurkan |
| 5 | Fork Node |  | satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

1. Sequence Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama | Simbol | Fungsi |
| 1 | Life Line,Objek Entity |  | antarmuka yang saling berinteraksi |
| 2 | Message |  | spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

1. Class Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama | Simbol | Fungsi |
| 1 | Generalization |  | hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor) |
| 2 | Class |  | himpunan dari objek-objek yang berbagi atribut serta operasi yang sama |
| 3 | Collaboration |  | deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 4 | Realization |  | operasi yang benar-benar dilakukan oleh suatu objek |
| 5 | Dependecy |  | hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempegaruhi elemen yang bergantung padanya |
| 6 | Association |  | apa yang menghubungkan antara objek |

**VIII. Jenis-Jenis UML**

1. Diagram Use Case(*Use Case Diagram)*  
   1.1 Pengertian Use Case Diagram

**Use Case Diagram adalah** pemodelan untuk menggambarkan behavior / kelakuan sistem yang akan dibuat. Use case diagram menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara sederhana, diagram use case digunakan untuk memahami fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang dapat menggunakan fungsi-fungsi tersebut.

Menurut Rosa dan Salahudin use case digram tidak menjelaskan secara detail tentang penggunaan tiap use case, namun hanya memberi gambaran singkat hubungan antara use case, aktor, dan sistem. Melalui use case diagram kita dapat mengetahui fungsi-fungsi apa saja yang ada pada sistem (Rosa-Salahudin, 2011: 130).

1.2 Deskripsi Use Case Diagram  
>>Diagram  use case merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem yang akan dibuat.

>>Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.

>>Diagram use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Yang ditekankan pada diagram ini adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

>>Sebuah use case merepresentasikan sebuah interaksi antara aktor (user atau sistem lainya) dengan sistem.

>>Use case menjelaskan secara sederhana fungsi sistem dari sudut pandang user.

1.3 Komponen-komponen yang terlibat dalam Use Case Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama | Simbol | Fungsi |
| 1 | Actor |  | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case* |
| 2 | Dependecy |  | hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan memperngaruhi elemen yang bergantung padanya elemen yang tidak mandiri(*independet*) |
| 3 | Generalization |  | hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor). |
| 4 | Include |  | menspesifikasikan bahwa use case sumber secara eksplisit. |
| 5 | Association |  | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 6 | System |  | menspesifikasikan paket yang menampilkan sistem secara terbatas |
| 7 | Use Case |  | deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 8 | Collaboration |  | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi) |
| 9 | Note |  | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi. |