

BAB II TINJAUAN PUSTAKA

2.1. Parkir

Parkir adalah keadaan tidak bergerak suatu kendaraan yang tidak bersifat sementara. Berhenti adalah keadaan tidak bergerak suatu kendaraan untuk sementara dengan pengemudi meninggalkan kendaraan. Fasilitas parkir adalah lokasi yang ditentukan sebagai tempat pemberhentian kendaraan yang tidak bersifat sementara untuk melakukan kegiatan pada suatu kurun waktu tertentu (Zaimuddin et al., 2019)

Menurut Undang – Undang Republik Indonesia Nomor : 22 tahun 2009 tentang Lalu Lintas dan Angkutan Jalan disebutkan bahwa Parkir adalah : Keadaan kendaraan berhenti atau tidak bergerak untuk beberapa saat dan ditinggal oleh pengemudinya. Pengelolaan parkir di Indonesia diserahkan pengelolaanya sepenuhnya ke pemerintah daerah, sehingga dimungkinkan retribusi parkir di suatu daerah akan berbeda dengan daerah lainnya.

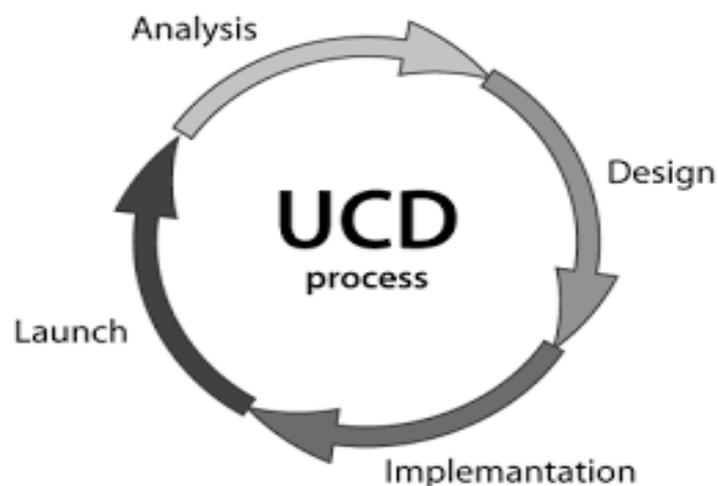
Parkir adalah istilah untuk menunjukkan suatu keadaan dimana kendaraan ditinggal sementara oleh pengendaranya di suatu lokasi tempat atau jalan. Semakin mudahnya syarat kepemilikan kendaraan bermotor membuat pemilik kendaraan bermotor tiap tahunnya meningkat pesat. Banyaknya leasing kendaraan bermotor yang memberi kemudahan dan memberi keringanan pembayaran secara dicicil pada siapapun yang ingin memiliki kendaraan bermotor membuat satu orang dapat memiliki kendaraan bermotor lebih dari satu (Handayani, 2018).

2.2. Sistem Parkir

Di Indonesia sistem parkir dikategorikan menjadi dua jenis yaitu sistem parkir di badan jalan (on-street parking) dan sistem parkir di dalam pelataran parkir (off-street parking). Pada sistem didalam pelataran parkir dikategorikan lagi menjadi dua jenis yaitu gedung parkir (parking building) dan parkir dibawah tanah (basement parking), berdasarkan pengaturan posisi kendaraan yaitu paralel, parkir tegak lurus, dan parkir serong (Baihaqi & Setiyadi, 2016).

2.3. *User Centered Design (UCD)*

UCD (*User Centered Design*) adalah sebuah pendekatan pembangunan perangkat lunak yang memusatkan fokus pada pengguna pada setiap proses pembangunan perangkat lunak. Pada UCD perancangan harus terkoneksi langsung dengan user/pengguna akhir atau calon user/pengguna melalui beberapa metode antara lain: wawancara, survey dan workshop pada saat perancangan. Pendekatan UCD dapat digunakan untuk meningkatkan usability dari produk. Akan tetapi, secara praktis penerapan UCD seringkali menemui berbagai kendala yang kurang memenuhi pencapaian user experience yang maksimal, kurang melibatkan end-to-end user pada tahap pengembangan, serta kurang menelusuri kepuasan pelanggan (T et al., 2015).



Gambar 2.1 Model UCD

Pada gambar 2.1 tersebut, Eason menggambarkan empat langkah utama dalam pengembangan, yakni perencanaan, perancangan, implementasi dan pengelolaan sistem. Tahap empat pendekatan yang bisa dijelaskan dari gambar diatas adalah :

1. *Soft System Methodology (SSM)* SSM lebih fokus pada perencanaan.
2. *Open Task Analysis (OSTA)*, OSTA lebih fokus pada langkah awal perancangan.
3. *Multiview* Metodologi yang lengkap dengan rentang perencanaan sampai dengan implementasi.
4. *Star Life Cycle* fokus utama pada perancangan.

2.4. Metode Template Matching

Menurut Muchtar and Apriadi, (2019) *Template Matching* adalah sebuah teknik dalam pengolahan citra digital untuk menemukan bagian – bagian terkecil dalam pencocokan gambar dengan *template* gambar sebagai acuan. *Template* dalam konteks rekognisi pola yang menunjuk pada konstruk internal yang jika cocok (*match*) dengan stimulus penginderaan mengantar pada rekognisi suatu objek. Untuk dapat bekerja *template matching* membutuhkan dua komponen utama, yaitu :

1. Sumber gambar: yaitu gambar yang dimana kita harapkan dapat menemukan kecocokan dengan gambar *template*.
2. Gambar *template*: yaitu gambar yang dijadikan sebagai *template* atau acuan yang akan dibandingkan dengan sumber gambar.

2.5. Bahasa Pemodelan Pengembangan Sistem (UML)

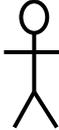
Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Rosa & Shalahuddin, 2018). Beberapa jenis diagram *UML* antara lain sebagai berikut:

2.5.1. Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat (Rosa & Shalahuddin, 2018). simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.1 di bawah ini:

Tabel 2.1 Simbol Diagram Use Case

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
Aktor/ <i>actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi

	walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i>
Ekstensi/ <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan
<< <i>extend</i> >> 	dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek biasanya <i>use case</i> tambahan memiliki nama depan
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/ <i>Include/uses</i> << <i>include</i> >> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya

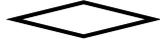
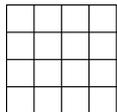
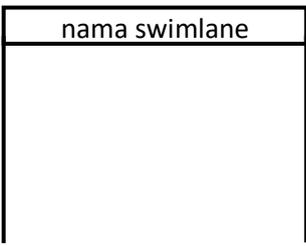
Sumber: (Rosa & Shalahuddin, 2018).

2.5.2. Activity Diagram

Activity diagram atau Diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa & Shalahuddin, 2018). simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2.2 Simbol Activity Diagram

Simbol	Deskripsi
Status awal	Status awal aktivitas sistem, sebuah diagram

	<p>aktivitas memiliki sebuah status awal.</p>
<p>Aktivitas</p> 	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
<p>Percabangan/<i>decision</i></p> 	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu</p>
<p>Penggabungan/<i>join</i></p> 	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu</p>
<p>Tabel</p> 	<p>Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis</p>
<p>Dokumen</p> 	<p>Menunjukkan dokumen sumber atau laporan</p>
<p>Status akhir</p> 	<p>Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.</p>
<p><i>Swimlane</i></p> 	<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

Sumber: (Rosa & Shalahuddin, 2018).

2.6. Pengertian *Mobile*

Mobile dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya *telephone mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA (Yuhefizar, 2014). Adapun karakteristik perangkat *mobile* yaitu:

1. Ukuran yang kecil: Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. *Memory* yang terbatas: Perangkat *mobile* juga memiliki *memory* yang kecil, yaitu *primary* (RAM) dan *secondary* (disk).
3. Daya proses yang terbatas: Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.
4. Mengonsumsi daya yang rendah: Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin *desktop*.
5. Kuat dan dapat diandalkan: Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas: Perangkat *mobile* memiliki *bandwith* rendah, beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek: Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

2.7. OpenCV

Menurut Muchtar and Apriadi, (2019) *OpenCV (Open Source Computer Vision)* merupakan perpustakaan computer source visi dan juga perangkat lunak pembelajaran. *OpenCV* dibangun untuk menyediakan insfrastruktur umum untuk aplikasi visi computer dan mempercepat penggunaan persepsi mesin dalam produk komersial. Kode library *OpenCV* memiliki lebih dari 2500 algortima yang

dapat dioptimalkan, yang mencakup satu set lengkap visi computer klasik dan mesin pembelajaran algoritma.

2.8. Anaconda

Anaconda menurut Wahyudin et al., (2019) yaitu Platform yang menyediakan paket distribusi Python. Kelebihan dari Anaconda data platform ini adalah semua library dapat dengan mudah melakukan konfigurasi yang berhubungan dengan modelling yang di kerjakan dalam satu tempat, semua environment nya sesuai dengan kebutuhan pengguna, dan anaconda juga menyediakan akses tools ke aplikasi lainnya seperti Jupyter Notebook, JupyterLab, Spyder, dan Microsoft Visual Studio Code.

2.9. Jupyter Notebook

Menurut Makasudede (2020) *Jupyter Notebook* adalah dokumen yang dihasilkan oleh *Jupyter Notebook App* yang berisikan kode komputer dan *rich text element* seperti paragraf, persamaan matematik, gambar dan tautan (*links*). *Jupyter Notebook* dahulu dikenal sebagai *IPython Notebook* dan kemudian akan berevolusi menjadi *Jupyter Lab*. Pada penelitian ini jupyter notebook digunakan untuk membangun rancangan kode program berbentuk *console*.

2.10. Bahasa Pemrograman Python

Menurut Makasudede (2020) Python adalah bahasa pemrograman yang bersifat umum. Karakteristik Python yang membuat python banyak disukai pengembang yaitu bahasa levelnya tinggi, stabil, dinamis, orientasi objek dan *cross platform*, tidak seperti bahasa yang lain yang sulit dibaca dan dipahami. Python lebih menekankan pada kode keterbacaan agar lebih mudah untuk memahami sintaks. Dimana hal ini yang membuat python mudah dipelajari baik yang pemula maupun yang sudah mahir menguasai bahasa pemrograman lain. Bahasa pemrograman python bisa dijalankan dikebanyakan hardware dan juga system operasi, sehingga kebanyakan komputer bisa menjalankannya.

Menurut Ayu & Utami (2021) Python adalah bahasa pemrograman yang fleksibel dan sederhana yang didefinisikan dalam dokumen-dokumennya sebagai

berikut. Python adalah sebuah bahasa pemrograman dinamis yang sering digunakan dalam pengembangan aplikasi pada berbagai domain. Hal ini memungkinkan suatu program ditulis dalam beberapa pendekatan sekaligus. Misalnya, antar muka grafis dibuat dalam bentuk orientasi objek, sedangkan pemrosesan dalam bentuk fungsional atau prosedural. Bahasa pemrograman Python memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak.

2.11. Penelitian Terkait

Berikut ini merupakan beberapa Penelitian yang berhubungan dengan laporan skripsi ini yaitu:

Table 2.3 Penelitian Terkait

No	Nama	Judul	Metode	Hasil
1	Budiman and Triono (2016)	Sistem Informasi Parkir Kendaraan Bermotor	Waterfall	Aplikasi berbasis mobile dengan yang dapat dioperasikan pada smartphone berbasis android. Sehingga manajemen perparkiran di lingkungan Universitas Merdeka Madiun dapat lebih efektif dan efisien
2	Zaimuddin <i>et al.</i> (2019)	Sistem Booking Parkir Mall Dengan Identifikasi Plat Nomor Kendaraan Berbasis	Prototype	Sistem booking parkir yang dibuat akan memberikan kemudahan dalam mendapatkan

		Android		tempat parkir saat pergi ke mall. Dengan system identifikasi plat nomor kendaraan dan digabungkan dengan system Android pelanggan mall akan dengan mudah mendapatkan tempat parkir.
3	Handayani (2018)	Rancang Bangun Sistem Parkir Dengan Kartu Barcode	Prototype	Pengelolaan parkir menggunakan kartu ini, pemilik kendaraan bermotor mendapat kemudahan dalam memarkirkan kendaraannya (khususnya roda 2). Dengan kartu ini juga diharapkan keamanan dalam area parkir lebih terjamin, dan area parkir terkelola dengan tertib