Yes, it's possible to use AI to assist in generating documentation from legacy PL/SQL code, although it's not a fully automated, "push-button" solution. The success depends heavily on the quality of the legacy code itself.

Here's how AI can help:

* **Code understanding and summarization:** AI models can analyze the PL/SQL code, identifying procedures, functions, packages, variables, and their relationships. They can then generate summaries of what each component does, based on the code's logic and (ideally) existing comments. Large language models (LLMs) excel at this type of natural language generation from code.
* **Comment extraction and improvement:** AI can extract existing comments from the code. It might even suggest improvements to incomplete or unclear comments.
* **Identifying potential issues:** The AI might highlight areas of the code that are particularly complex, poorly documented, or potentially problematic (e.g., hardcoded values, lack of error handling). This helps focus documentation efforts where they're most needed.
* **Generating documentation templates:** AI can help create templates for different types of documentation, such as API documentation or user manuals, based on the information it extracts from the code.

**Limitations:**

* **Poorly commented code:** If the legacy PL/SQL code is poorly commented or lacks meaningful variable names, the AI's ability to generate accurate and useful documentation will be severely limited. The AI is learning from the code; poor code leads to poor documentation.
* **Complex logic:** Extremely complex algorithms or intricate data flows might be difficult for the AI to fully understand and summarize accurately. Human review and interpretation will be necessary.
* **Ambiguous code:** The AI may struggle with ambiguous or poorly written code, leading to potentially inaccurate documentation.
* **Data model understanding:** The AI may not have a complete understanding of the data model the PL/SQL code interacts with, which can impact the accuracy of the generated documentation.

**How to use AI for this task:**

1. **Code cleanup:** Before using AI, consider cleaning up the legacy code as much as possible. This includes refactoring, improving variable names, and adding or improving existing comments.
2. **Choose an appropriate tool:** Several tools integrate AI capabilities for code analysis and documentation generation. Some might even have plugins or extensions specifically for PL/SQL (although this is an area that's constantly evolving). Explore options like Github Copilot, other LLMs, or specialized code analysis platforms.
3. **Iterative process:** Expect this to be an iterative process. The AI will generate a first draft; you'll need to review, edit, and refine the generated documentation to ensure accuracy and completeness.

In short, AI can be a valuable assistant, but it's not a replacement for human expertise when documenting legacy PL/SQL code. Consider it a powerful tool to accelerate the process, not a fully automated solution.